SUN SEEBEYOND

eWAY™ ADAPTER FOR MSMQ USER'S GUIDE

Release 5.1.2

Copyright © 2006 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved. Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at http://www.sun.com/patents and one or more additional patents or pending patent applications in the U.S. and in other countries. U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements. Use is subject to license terms. This distribution may include materials developed by third parties. Sun, Sun Microsystems, the Sun logo, Java, Sun Java Composite Application Platform Suite, SeeBeyond, eGate, eInsight, eVision, eTL, eXchange, eView, eIndex, eBAM, eWay, and JMS are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon architecture developed by Sun Microsystems, Inc. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd. This product is covered and controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

Copyright © 2006 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés. Sun Microsystems, Inc. détient les droits de propriété intellectuels relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuels peuvent inclure un ou plus des brevets américains listés à l'adresse http://www.sun.com/patents et un ou les brevets supplémentaires ou les applications de brevet en attente aux Etats - Unis et dans les autres pays. L'utilisation est soumise aux termes de la Licence. Cette distribution peut comprendre des composants développés par des tierces parties. Sun, Sun Microsystems, le logo Sun, Java, Sun Java Composite Application Platform Suite, Sun, SeeBeyond, eGate, eInsight, eVision, eTL, eXchange, eView, eIndex, eBAM et eWay sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd. Ce produit est couvert à la législation américaine en matière de contrôle des exportations et peut être soumis à la règlementation en vigueur dans d'autres pays dans le domaine des exportations et importations. Les utilisations, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes biologiques et chimiques ou du nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers les pays sous embargo américain, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exhaustive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la législation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement désignés, sont rigoureusement interdites.

Part Number: 819-7407-10

Version 20061004090713

Contents

Chapter 1

Introduction 6

About Microsoft™ Message Queuing (MSMQ) 6

About the Sun SeeBeyond eWay™ Adapter for MSMQ 7

What's New in This Release 7

What's in This Document 8

MSMQ Javadoc 8

Scope of the Document 8

Intended Audience 8

Text Conventions 8

Sun Microsystems, Inc. Web Site 9

Documentation Feedback 9

Chapter 2

Installing the MSMQ eWay 10

MSMQ eWay System Requirements 10

Installing the MSMQ eWay 10

Installing the eWay on a JavaCAPS Supported System 11

Adding the eWay to an Existing Suite Installation 11

Installing eWay Enterprise Manager plug-ins 12

MSMQ eWay Alert Codes 12

After Installation 14

Downloading and Adding the MSMQ DLLs and JNI files 14

Adding the Runtime JNI File 15

Adding the DLL Files to the Environment Path 15

ICAN 5.0 Project Migration Procedures 15

Chapter 3

Configuring the MSMQ eWay 18

Creating and Configuring the MSMQ eWay 18

Selecting MSMQ as the External Application 18

Modifying the MSMQ eWay Properties 19
Using the Properties Editor 20
MSMQ eWay Properties 21
Inbound MSMQ eWay Connectivity Map Properties 22
MSMQ Format Name and Host Name 25
Outbound MSMQ eWay Connectivity Map Properties 27
MSMQ eWay Environment Properties 32
Inbound MSMQ eWay > MSMQ Environment 32
Inbound MSMQ eWay > MDB Settings 33
Inbound MSMQ eWay > Connection Retry Settings 34
Outbound MSMQ eWay > MSMQ Environment 35
Outbound MSMQ eWay > Connection Retry Settings 36
Enabling Rollback When an MSMQ Message Fails 37
Manually Configuring the eWay 38

Chapter 4
Implementing an MSMQ eWay Project using eInsight 39
eInsight Engine and Components 39
The MSMQ eWay With eInsight 40
MSMQ eWay Sample Project for eInsight 40
Importing a Sample Project 41
Creating the prjMSMQBPM Project 41
Creating a Project 41
Creating OTDs Using the DTD Wizard 42
Creating the Business Processes 43
Creating the bpGet Business Process 43
Creating the Business Rules Using the Business Rules Designer 45
Creating the bpGet Business Process Business Rules 45
Creating the bpPut Business Process Business Rules 48
Creating the bpReceive Business Process Business Rules 50
Creating the Connectivity Maps 53
Binding the Project Components 55
Binding the components of the cmPut Connectivity Map 56
Binding the components of the cmPut Connectivity Map 56
Creating an Environment 56
Configuring the eWays 57
cmGet Connectivity Map Properties 57
cmPut Connectivity Map Properties 57
cmReceive Connectivity Map Properties 58
Environment Properties 58
Creating and Activating the Deployment Profile 58
Configuring the Integration Server 59
Creating and Starting the Domain 59
Building and Deploying the Project 60
Running the Sample 60

Chapter 5

Implementing an MSMQ eWay Project using Java Collaboration Definitions 61

MSMQ eWay Components 61

MSMQ eWay Sample Project 62

Importing a Sample Project 62

Creating the prjMSMQJCD Project 63

- Creating a New Project 63
- Creating OTDs Using the DTD Wizard 64
- Creating the Collaboration Definitions 64
  - Create the jcdGet Collaboration 65
  - Create the jcdPut Collaboration 66
  - Create the jcdReceive Collaboration 67
- Creating the Business Rules Using the Collaboration Editor 67
  - Creating the jcdGet Collaboration Business Rules 67
  - Creating the jcdPut Collaboration Business Rules 73
  - Creating the jcdReceive Collaboration Business Rules 77
- Creating the Connectivity Maps 80
- Binding the Project Components 82
  - Binding the components of the cmPut Connectivity Map 83
  - Binding the components of the cmPut Connectivity Map 83
- Creating an Environment 84
- Configuring the eWays 84
  - **cmGet Connectivity Map Properties 85**
  - **cmPut Connectivity Map Properties 85**
  - **cmReceive Connectivity Map Properties 85**
  - Environment Properties 85
- Creating the Deployment Profile 86
- Configuring the Integration Server 87
- Creating and Starting the Domain 87
- Building and Deploying the Project 88
- Running the Sample 88

Index 89

Chapter 1

Introduction

Welcome to the Sun SeeBeyond eWay™ Adapter for MSMQ User's Guide. This document describes how to install, configure, and implement the Sun SeeBeyond eWay™ Adapter for MSMQ in a typical Sun Java™ Composite Application Platform Suite environment.

What's in This Chapter

- **About Microsoft™ Message Queuing (MSMQ)** on page 6
- **About the Sun SeeBeyond eWay™ Adapter for MSMQ** on page 7
- **What's in This Document** on page 8
- **Sun Microsystems, Inc. Web Site** on page 9

1.1 About Microsoft™ Message Queuing (MSMQ)

Microsoft Message Queuing, or MSMQ, is message queuing communications "middleware" that enables applications on diverse systems to communicate with each other. MSMQ allows applications to communicate through messages, and can guarantee the delivery of messages even in the event of system or network failures.

By using messages, MSMQ can provide asynchronous communication between applications running on the same computer or on separate computers connected through a network. When an application receives a message, it processes the message by reading the message content and acting accordingly. If appropriate, the receiving application can send a response message back to the sending application. MSMQ protects these transactions by safely storing the messages in queues, which prevents message loss and make messages available to an application whenever it is prepared to receive them.

Applications using MSMQ are able to send and process messages, even when the receiving application is not available. Upon connection, the receiver can accept or pick up the messages. Applications using MSMQ's transactional delivery mode can be assured that messages are delivered exactly one time only, and that they are delivered in the order that they were sent.

1.2 About the Sun SeeBeyond eWay™ Adapter for MSMQ

The Sun SeeBeyond eWay™ Adapter for MSMQ (referred to as the MSMQ eWay throughout this document) allows the Sun Java™ Composite Application Platform Suite to exchange data with Microsoft's MSMQ. The MSMQ eWay transparently integrates existing systems with MSMQ.

The MSMQ eWay enables the Sun Java™ Composite Application Platform Suite to use business logic within Collaborations or Business Processes to perform data identification, manipulation, and transformation. Messages are tailored to meet the communication requirements of specific applications or protocols. This document explains how to install and configure the MSMQ eWay.

1.3 What's New in This Release

The Sun SeeBeyond eWay™ Adapter for MSMQ includes the following changes and new features:

New for Version 5.1.2

- This is a maintenance release. No new features.

New for Version 5.1.1

- Version Control: An enhanced version control system allows you to effectively manage changes to the eWay components.
- Manual Connection Management: Establishing a connection can now be performed automatically (configured as a property) or manually (using OTD methods from the Java Collaboration).
- Multiple Drag-and-Drop Component Mapping from the Deployment Editor: The Deployment Editor now allows you to select multiple components from the Editor's component pane, and drop them into your Environment component.
- Support to read configuration parameters from LDAP at runtime.
- Connection Retry Support: Allows you to specify the number of attempts to reconnect, and the interval between retry attempts, in the event of a connection failure.
- Connectivity Map Generator: Generates and links your Project's Connectivity Map components using a Collaboration or Business Process.

Many of these features are documented further in the *Sun SeeBeyond eGate Integrator User's Guide* or the *Sun SeeBeyond eGate Integrator System Administrator Guide*.

1.4 What's in This Document

This document provides information about installing, configuring, and using the Sun SeeBeyond eWay™ Adapter for MSMQ and includes the following chapters:

- **Chapter 1 "Introduction"** provides an overview of the MSMQ eWay.
- **Chapter 2 "Installing the MSMQ eWay"** describes how to install the MSMQ eWay and lists the supported operating systems and system requirements.
- **Chapter 3 "Configuring the MSMQ eWay"** describes the process of configuring the MSMQ eWay to run in your environment.
- **Chapter 4 "Implementing an MSMQ eWay Project using eInsight"** describes the features and functionality of the MSMQ eWay using the eInsight Business Process Manager and the eInsight Web Services interface.
- **Chapter 5 "Implementing an MSMQ eWay Project using Java Collaboration Definitions"** describes the features and functionality of the MSMQ eWay using the eGate Integrator and the Collaboration Editor (Java).

MSMQ Javadoc

An MSMQ eWay Javadoc is also provided, which documents the Java methods that are available with the MSMQ eWay. The Javadoc is uploaded with the eWay's documentation file (MSMQeWayDocs.sar) and downloaded from the Documentation tab of the Sun Java™ Composite Application Platform Suite Installer. To access the full Javadoc, extract the Javadoc to an easily accessible folder, and double-click the index.html file.

1.4.1 Scope of the Document

This user's guide provides a description of the Sun SeeBeyond eWay™ Adapter for MSMQ. It includes directions for installing the eWay, configuring the eWay properties, and implementing the eWay's sample Projects. This document is also intended as a reference guide, listing available properties, functions, and considerations. For a reference of available MSMQ eWay Java methods, see the associated Javadoc.

1.4.2 Intended Audience

This guide is intended for experienced computer users who have the responsibility of helping to set up and maintain a fully functioning Java Composite Application Platform Suite system. This person must also understand any operating systems on which the Java Composite Application Platform Suite will be installed (Windows and UNIX), and must be thoroughly familiar with Windows-style GUI operations.

1.4.3 Text Conventions

The following conventions are observed throughout this document.

Table 1 Text Conventions

Text Convention	Used For	Examples
Bold	Names of buttons, files, icons, parameters, variables, methods, menus, and objects	▪ Click **OK**. ▪ On the **File** menu, click **Exit**. ▪ Select the **eGate.sar** file.
`Monospaced`	Command line arguments, code samples; variables are shown in ***bold italic***	`java -jar filename.jar`
Blue bold	Hypertext links within document	See **Text Conventions** on page 8
Blue underlined	Hypertext links for Web addresses (URLs) or email addresses	http://www.sun.com

1.5 Sun Microsystems, Inc. Web Site

The Sun Microsystems web site is your best source for up-to-the-minute product news and technical support information. The site's URL is:

http://www.sun.com

1.6 Documentation Feedback

We appreciate your feedback. Please send any comments or suggestions regarding this document to:

CAPS_docsfeedback@sun.com

Chapter 2

Installing the MSMQ eWay

This chapter describes the process of installing the MSMQ eWay and the accompanying sample Projects.

What's in This Document

- **Installing the MSMQ eWay** on page 10
- **Downloading and Adding the MSMQ DLLs and JNI files** on page 14
- **ICAN 5.0 Project Migration Procedures** on page 15

2.1 MSMQ eWay System Requirements

The MSMQ eWay Readme contains the latest information on:

- Supported Operating Systems
- System Requirements
- External System Requirements
- The MSMQ eWay Readme is uploaded with the eWay's documentation file (MSMQeWayDocs.sar) and can be accessed from the Documentation tab of the Sun Java™ Composite Application Platform Suite Installer. Refer to the MSMQ eWay Readme for the latest requirements before installing the MSMQ eWay.

2.2 Installing the MSMQ eWay

The Sun Java™ Composite Application Platform Suite Installer, a web-based application, is used to select and upload eWays and add-on files during the installation process. The following section describes how to install the components required for this eWay.

2.2.1 Installing the eWay on a JavaCAPS Supported System

Follow the directions for installing the Sun Java™ Composite Application Platform Suite in the *Sun Java™ Composite Application Platform Suite Installation Guide. After you have installed eGate or eInsight, do the following:*

1. From the Installer's **Select Sun Java™ Composite Application Platform Suite Products to Install** table (Administration tab), expand the **eWay** option.
2. Select the products for your Suite and include the following:
   - **FileeWay** (the File eWay is used by most sample Projects)
   - **MSMQeWay**

   To upload the MSMQ eWay User's Guide, Help file, Javadoc, Readme, and sample Projects, select the following:
   - **MSMQeWayDocs**
3. Once you have selected all of your products, click **Next** in the top-right or bottom-right corner of the **Select Sun Java™ Composite Application Platform Suite Products to Install** box.
4. From the **Selecting Files to Install** box, locate and select your first product's SAR file. Once you have selected the SAR file, click **Next**. Your next selected product appears. Follow this procedure for each of your selected products. The **Installation Status** window appears and installation begins after the last SAR file has been selected.
5. Once your product's installation is finished, continue installing the Sun Java™ Composite Application Platform Suite as instructed in the *Sun Java™ Composite Application Platform Suite Installation Guide.*

The MSMQ eWay also requires additional runtime **DLL** and **JNI** files that are included with the installation. For directions on how to install these additional files see **"Downloading and Adding the MSMQ DLLs and JNI files" on page 14**

Adding the eWay to an Existing Suite Installation

If you are adding the eWay to an existing Sun Java™ Composite Application Platform Suite installation, do the following:

1. Complete steps 1 through 4 above.
2. Once your product's installation is finished, open the Enterprise Designer and select **Update Center** from the Tools menu. The **Update Center Wizard** appears.
3. For Step 1 of the wizard, simply click **Next**.
4. For Step 2 of the wizard, click the **Add All** button to move all installable files to the **Include in Install** field, then click **Next**.
5. For Step 3 of the wizard, wait for the modules to download, then click **Next**.
6. The wizard's Step 4 window displays the installed modules. Review the installed modules and click **Finish.**

7 When prompted, restart the IDE (Integrated Development Environment) to complete the installation.

2.2.2 Installing eWay Enterprise Manager plug-ins

The **Sun SeeBeyond Enterprise Manager** is a Web-based interface that allows you to monitor and manage your Suite applications. The Enterprise Manager requires an eWay specific "plug-in" for each of your installed eWays. These plug-ins enable the Enterprise Manager to target specific alert codes for each eWay type, as well as to start and stop the inbound eWays.

The *Sun Java™ Composite Application Platform Suite Installation Guide* describes how to install the Sun SeeBeyond Enterprise Manager. The *Sun SeeBeyond eGate™ Integrator System Administration Guide* describes how to monitor servers, Services, logs, and alerts using the Sun SeeBeyond Enterprise Manager and the command-line client.

The **eWay Enterprise Manager plug-ins** are available from the **List of Components to Download** under the Installer's **DOWNLOADS** tab.

There are two ways to add the eWay Enterprise Manager plug-ins:

1 From the Enterprise Manager:

A From the **Enterprise Manager**'s Explorer toolbar, click the **Configuration** icon.

B Click the **Web Applications Manager** tab, go to the **Auto-Install from Repository** tab, and connect to your Repository.

C Select the application plug-ins you require, and click **Install**. The application plug-ins are installed and deployed.

2 From the **Sun Java™ Composite Application Platform Suite Installer:**

A From the Installer's **Download** tab, select the Plug-Ins you require and save them to a temporary directory.

B Log onto the **Enterprise Manager**. From the **Enterprise Manager**'s Explorer toolbar, click the **Configuration** icon.

C Click the **Web Applications Manager** tab and go to the **Manage Applications** tab.

D Browse for and select the WAR file for the application plug-in that you downloaded, and click **Deploy**. The plug-in is installed and deployed.

MSMQ eWay Alert Codes

You can view and delete alerts using the Enterprise Manager. An alert is triggered when a specified condition occurs in a Project component. The purpose of the alert is to warn the administrator or user that a condition has occurred.

To View the eWay Alert Codes

1 Add the eWay Enterprise Manager plug-in for this eWay.

2 From the Enterprise Manager's **Explorer** toolbar, click the **Configuration** icon.

3 Click the **Web Applications Manager** tab and go to the **Manage Alert Codes** tab. Your installed alert codes are displayed under the **Results** section. If your eWay alert codes are not available displayed under **Results**, do the following

A From the **Install New Alert Codes** section, browse to and select the eWay alert properties file for the application plug-in that you added. The alert properties files are located in the **alertcodes** folder of your Sun Java Composite Application Platform Suite installation directory.

B Click **Deploy**. The available alert codes for your application are displayed under **Results**. A listing of available this eWay's alert codes is displayed in Table 2.

Table 2 MSMQ eWay Alert Codes

Alert Code	Description	User Action
MSMQ-CLOSEQUEUEFAILED	Failed to close MSMQ Queue.	▪ Restart MSMQ. ▪ Refer to alert details and server log file for more details.
MSMQ-CONNECTIONFAILED	Failed to establish connection to MSMQ server.	▪ Verify your eWay configuration properties. ▪ Verify that your MSMQ server is running.
MSMQ-DISCONNECTIONFAILED	Failed to disconnect from MSMQ server: {0}.	▪ Restart MSMQ. ▪ Refer to alert details and server log for more details.
MSMQ-OPENQUEUEFAILED	Failed to open MSMQ Queue: {0}.	▪ Verify your eWay configuration properties. ▪ Verify that your MSMQ server is running.
MSMQ-RECEIVEMESSAGEFAILED	Failed to receive message from MSMQ Queue: {0}.	▪ Verify that your MSMQ server and MSMQ service is running. ▪ Refer to alert details and server log for the MSMQ API error code.
MSMQ-SENDMESSAGEFAILED	Failed to send message to MSMQ Queue: {0}.	▪ Verify that your MSMQ server and MSMQ service is running. ▪ Refer to alert details and server log for the MSMQ API error code.
MSMQ-GETMESSAGEFAILED	Failed to get message from MSMQ Queue: {0}.	▪ Verify that your MSMQ server and MSMQ service is running. ▪ Refer to alert details and server log for the MSMQ API error code.
MSMQEWAY-SUSPENDED	MSMQ eWay is suspended.	▪ eWay is no longer running. Restart the eWay.

An alert code is a warning that an error has occurred. It is not a diagnostic. The user actions noted above are just some possible corrective measures you may take. Refer to the log files for more information. For information on managing and monitoring alert codes and logs, see the *Sun SeeBeyond eGate Integrator System Administration Guide.*

2.2.3 After Installation

Once the eWay is installed and configured, you must then incorporate it into a Project before it can perform its intended functions. See the *Sun SeeBeyond eGate Integrator User's Guide* for more information on incorporating the eWay into an eGate Project.

The MSMQ eWay User's Guide, Javadoc, Readme, and sample Projects, can be accessed or downloaded from the Sun Java™ Composite Application Platform Suite Installer's Documentation tab.

2.3 Downloading and Adding the MSMQ DLLs and JNI files

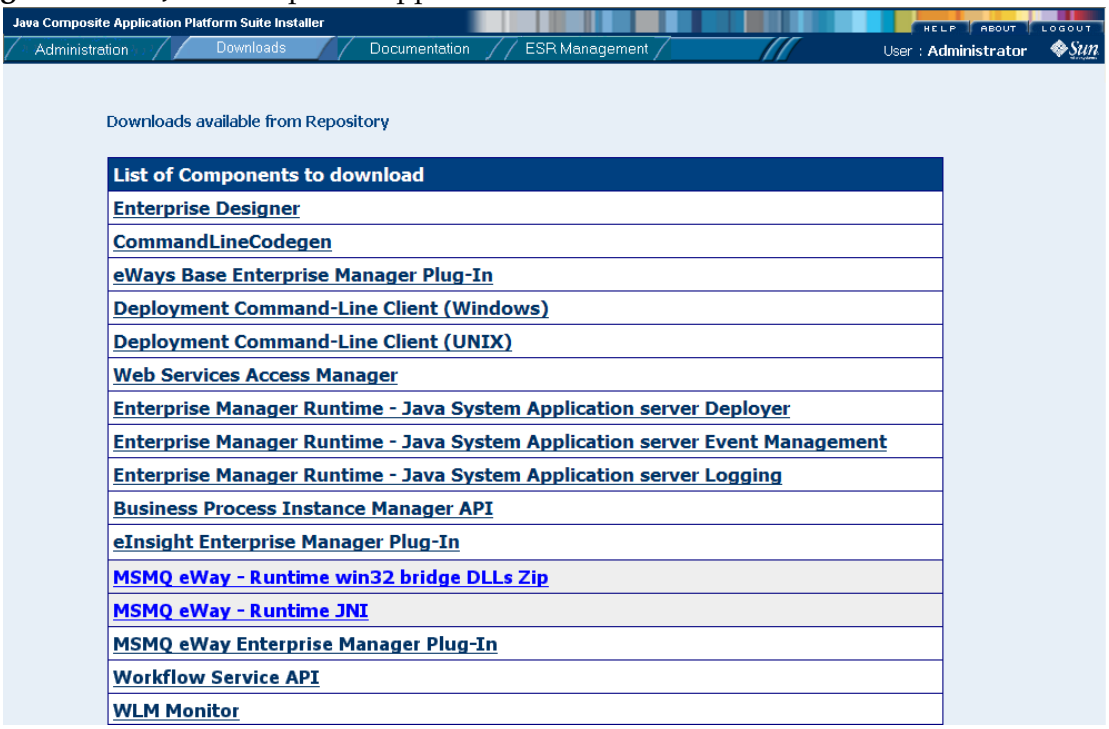
The MSMQ eWay installation includes two additional components (as well as the Enterprise Manager Plug-In), as see in Figure 1, that must be downloaded and installed for the MSMQ eWay:

- **MSMQ eWay - Runtime win32 bridge DLLS Zip**: This file contains a number of required DDL files that must be copied to the Windows Environment Variable Path.
- **MSMQ eWay - Runtime JNI**: This file must be downloaded using the Sun Java™ Composite Application Platform Suite Installer.

To download the MSMQ DLLs and Runtime JNI do the following:

1. From the Sun Java™ Composite Application Platform Suite Installer, click the **DOWNLOADS** tab. The MSMQ DLLs and Runtime JNI Component list are available from the Repository Downloads list.

Figure 1 Sun Java Composite Application Platform Suite Installer - Downloads



2 Download both files to a local directory. For directions on adding the msmqjni.jar file to the Windows Environment path, see **"Adding the Runtime JNI File" on page 15**. For directions on adding the DLL files to the PATH for the Logical Host process, see **"Adding the DLL Files to the Environment Path" on page 15**.

2.3.1 Adding the Runtime JNI File

Prior to creating your MSMQ eWay Project, copy the **Runtime JNI (msmqjni.jar)**, downloaded from the Suite Installer, to the following locations:

```
<JavaCAPS51>\edesigner\lib\ext
<JavaCAPS51>\logicalhost\is\lib
```

where <*JavaCAPS51*> is the directory in which Sun Java™ Composite Application Platform Suite is installed.

These file must be copied manually.

Note: *You need to copy the msmqjni.jar file to the \compile\lib\ext folder before deploying and running command line codegen. You also need to copy the msmqjni.jar file to the c:\Sun\ApplicationServer\lib folder before deploying and running via the Sun Java™ System Application Server Enterprise Edition 8.1.*

2.3.2 Adding the DLL Files to the Environment Path

The Runtime win32 bridge DLL files must be added to a directory that is included in the Windows Environment System Variable Path before running an MSMQ Project. To make sure that the correct file is added to the proper path location, do the following:

1 Locate the **msmqruntimejni.zip** file that you downloaded to a local folder.

2 Extract the ZIP file to a directory that is part of the Windows Environment path.

2.4 ICAN 5.0 Project Migration Procedures

This section describes how to transfer your current ICAN 5.0 Projects to Sun Java Composite Application Platform Suite, version 5.1.2. Only Projects developed on ICAN 5.0.2 and above can be migrated successfully to the Sun Java Composite Application Platform Suite. To migrate your ICAN 5.0 Projects, do the following:

Export the Project

1 Before you export your Projects, save your current ICAN 5.0 Projects to your Repository.

2 From the Project Explorer, right-click your Repository and select **Export** from the shortcut menu. The Export Manager appears.

3 From the **Select Projects from the list field** of the Export Manager, select one or more Projects that you want to export and move them to the **Selected Projects** field by clicking the **Add to Select Items** (arrow) button, or click **All** to include all of your Projects.

4 In the same manner, from the **Select Environments from the list** field, select the Environments that you want to export and move them to the Selected Environments field by clicking the **Add to Select Items** (arrow) button, or click **All** to include all of your Environments.

5 Browse to select a destination for your Project ZIP file and enter a name for your Project in the **ZIP file** field.

6 Click **Export** to create the Project ZIP file in the selected destination.

Install Sun Java Composite Application Platform Suite

7 Install Sun Java Composite Application Platform Suite, including all eWays, libraries, and other components used by your ICAN 5.0 Projects.

8 Start the Sun SeeBeyond Enterprise Designer.

Import the Project

9 From the Enterprise Designer's Project Explorer, right-click the Repository and select **Import Project** from the shortcut menu. The Import Manager appears.

10 Browse to and select your exported Project file.

11 Click **Import**. A warning message, **"Missing APIs from Target Repository**," may appear at this time. This occurs because various product APIs were installed on the ICAN 5.0 Repository, when the Project was created, that are not installed on the Sun Java Composite Application Platform Suite Repository. These APIs may or may not apply to your Projects. You can ignore this message if you have already installed all of the components that correspond to your Projects. Click **Continue** to resume the Project import.

12 Close the Import Manager after the Project is successfully imported.

Deploy the Project

13 You must create a new Deployment Profile for each of your imported Projects. When you export a Project, the Project's components are automatically *"checked in"* to Version Control to write-protect each component. These protected components appear in the Explorer tree with a red padlock in the bottom-left corner of each icon. Before you can deploy the imported Project, the Project's components must first be *"checked out"* of Version Control from both the Project Explorer and the Environment Explorer. To *"check out"* all of the Project's components, do the following:

A From the Project Explorer, right-click the Project and select **Version Control > Check Out** from the shortcut menu. The Version Control - Check Out dialog box appears.

B Select **Recurse Project** to specify all components, and click **Check Out**.

C Select the Environment Explorer tab, and from the Environment Explorer, right-click the Project's Environment and select **Version Control > Check Out** from the shortcut menu.

D Select **Recurse Environment** to specify all components, and click **Check Out**.

14 If your imported Projects include File eWays, they must be reconfigured in your Environment prior to deploying the Project. To reconfigure your File eWays, do the following:

   A The Environment File External System properties can now accommodate both inbound and outbound eWays. If your previous Environment includes both inbound and outbound File External Systems, delete one of these (for example, the outbound File External System).

   B From the Environment Explorer tree, right-click your remaining File External System, and select **Properties** from the shortcut menu. The Properties Editor appears.

   C The Directory property has been relocated from the Connectivity Map Properties to the Environment Properties. Set the inbound and outbound Directory values, and click **OK**.

15 Deploy your Projects.

Note: *Only Projects developed on ICAN 5.0.2 and above can be imported and migrated successfully into the Sun Java Composite Application Platform Suite.*

Chapter 3

Configuring the MSMQ eWay

This chapter describes how to create and configure the MSMQ eWay.

What's in This Chapter

- **Using the Properties Editor** on page 20
- **Inbound MSMQ eWay Connectivity Map Properties** on page 22
- **Outbound MSMQ eWay Connectivity Map Properties** on page 27
- **MSMQ eWay Environment Properties** on page 32
- **Enabling Rollback When an MSMQ Message Fails** on page 37
- **Manually Configuring the eWay** on page 38

3.1 Creating and Configuring the MSMQ eWay

All eWays contain a set of properties that are unique to that eWay type. After the eWays are created and an MSMQ External System is added to the Project's Environment, the eWay parameters can be modified for your specific system. The MSMQ eWay properties are modified from two locations:

- From the **Connectivity Map**. These properties most commonly apply to a specific eWay, and may vary from other eWays (of the same type) in the Project. The MSMQ eWay's Connectivity Map properties vary depending on whether the eWay is an inbound or outbound eWay.
- From the **Environment Explorer tree**. These properties are commonly global, and apply to all eWays (of the same type) in the Project. The saved properties are shared by all eWays in the MSMQ External System window.

The properties for the MSMQ eWay must be set in both locations.

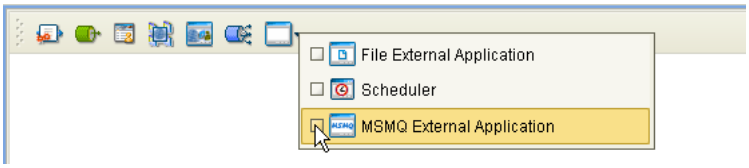
3.1.1 Selecting MSMQ as the External Application

To create an MSMQ eWay, you must first create an MSMQ External Application in your Connectivity Map. MSMQ eWays are located between an MSMQ External Application and a Service. Services are containers for Collaborations, Business Processes, eTL processes, and so forth.

To create the MSMQ External Application

1. From the Connectivity Map toolbar, click the **External Applications** icon.
2. Select the **MSMQ External Application** from the menu (see Figure 2). The selected MSMQ External Application icon appears on the Connectivity Map toolbar.

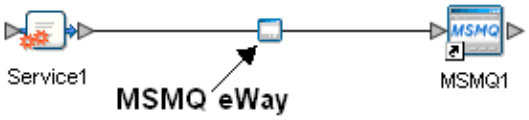
Figure 2 External Applications Selection Menu



3. Drag the new **MSMQ External Application** from the toolbar onto the Connectivity Map canvas. This represents an external MSMQ system.

From the Connectivity Map, you can associate (bind) the External Application to the Service to establish an eWay (see Figure 3).

Figure 3 eWay Location



When MSMQ is selected as the External Application, it automatically applies the default MSMQ eWay properties, provided by the OTD, to the eWay that connects it to the Service. You can then modify these properties for your specific system using the **Properties Editor**.

3.1.2 Modifying the MSMQ eWay Properties

A Project's eWay properties can be modified after the eWays have been created in the Connectivity Map and the External Applications have been added to the Project's Environment.

Modifying the MSMQ eWay (Connectivity Map) Properties

1. From the Connectivity Map, double-click the eWay icon located in the link between the associated External Application and the Service. The Templates dialog box appears.
2. From the Templates dialog box, select **Inbound** or **Outbound** as the eWay configuration type and click **OK**.
3. The eWay **Properties Editor** opens to the inbound or outbound MSMQ Connectivity Map properties. Make any necessary modifications and click **OK** to save the settings.

Modifying the MSMQ eWay (Environment Explorer) Properties

1. From the Environment Explorer tree, right-click the MSMQ External System. Select **Properties** from the shortcut menu. The **Properties Editor** appears.

2 Make any necessary modifications to the Environment parameters of the MSMQ eWays, and click **OK** to save the settings.

3.1.3 Using the Properties Editor

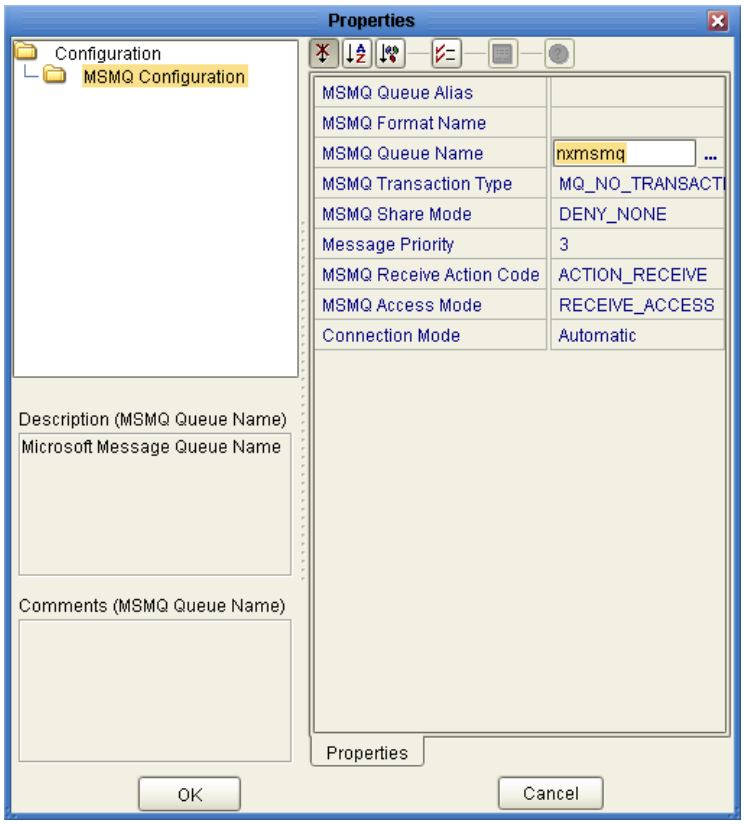
You can modify the default eWay configuration properties from the MSMQ eWay Properties Editor.

To modify the eWay properties do the following:

1 Open the MSMQ eWay Properties Editor for the eWay you want to edit. The MSMQ inbound and outbound eWays have two sets of parameters: those specific to that particular eWay (accessed from the **Connectivity Map**), and those that are common to all eWays of this type (accessed from the **Environment Explorer** tree).

2 From the upper-left pane of the Properties Editor, select a properties directory from the Configuration tree. The parameters contained in that directory are now displayed in the right pane of the Properties Editor. For example, from the outbound eWay Connectivity Map Properties, click on **MSMQ Configuration** to display this section's editable parameters in the right pane, as shown in Figure 4.

Figure 4 Properties Editor -- MSMQ Properties



3 Click on any property field to make it editable. For example, click on the **MSMQ Queue Name** parameter to edit the MSMQ Queue Name value.

If a parameter's value is true/false or multiple choice, the field, when selected,

reveals a submenu of property options. If a parameter requires that you type in a value, such as a name or password, the property field provides space to type in the value and an ellipsis (. . .) button.

Click on the ellipsis (. . .) in the properties field to open a separate configuration dialog box. This is helpful for entering large values that cannot be fully displayed in the parameter's property field. Enter the property value in the dialog box and click **OK**. The value is now displayed in the property field.

4 A description of each parameter is displayed in the **Description** pane when that parameter is selected, providing an explanation of any required settings or options.

5 The **Comments** pane provides an area for recording notes and information regarding the currently selected parameter. This is saved for future reference.

6 After modifying the configuration properties, click **OK** to close the Properties Editor and save the changes.

3.2 MSMQ eWay Properties

The MSMQ eWay's Properties are organized as follows:

Inbound MSMQ eWay Connectivity Map Properties on page 22

Outbound MSMQ eWay Connectivity Map Properties on page 27

MSMQ eWay Environment Properties on page 32

Note: *Modifying individual OTD configuration settings can override the default eWay OTD configuration settings.*

3.3 Inbound MSMQ eWay Connectivity Map Properties

The **MSMQ Configuration** section of the **inbound MSMQ** Connectivity Map properties contains the top-level parameters displayed in Table 3.

Identifying an MSMQ Queue

The MSMQ eWay identifies an MSMQ Queue using the first available value of these three properties, in the following order:

1. MSMQ Queue Alias
2. MSMQ Format Name
3. MSMQ Queue Name

When use one of these properties, for example **Format Name**, to identify a queue, we recommend that you leave the value for the other two properties blank (in this case, you would leave the **MSMQ Queue Alias** and **MSMQ Queue Name** properties blank). This ensures that only the queue identification method you specify is used.

Table 3 Connectivity Map - Inbound - MSMQ Configuration

Name	Description	Required Value
MSMQ Queue Alias	Specifies the queue alias. A queue alias associates an ADs path and a user-defined string (alias) with a public, private, or direct single-element format name. To send or receive messages using a queue alias, **MSMQ** and the **Active Directory** service must be installed on the same computer as the Logical Host.	The queue alias. See **"Identifying an MSMQ Queue" on page 22** for more information.

Table 3 Connectivity Map - Inbound - MSMQ Configuration (Continued)

Name	Description	Required Value
MSMQ Format Name	Specifies the format name of the queue. The format name is a string that uniquely identifies a queue by providing some connection details and the queue's path. Different types of format names can be used to specify the way messages are routed, the type of destination, and the type of operation for which the queue is being opened. Configure your MQ security attributes to accept TCP and HTTP protocol.	The public or private format name in the following manner: ▪ For **public** queues use the following: ♦ DIRECT=TCP:*IPAddress**QueueName* ♦ DIRECT=OS:*ComputerName**QueueName* ♦ DIRECT=HTTP://*Host*/msmq/*QueueName* ▪ For **private** queues use the following: ♦ DIRECT=TCP:*IPAddress*\private$*QueueName* ♦ *DIRECT=OS:ComputerName*\private$*Queue Name* Configure your MQ security attributes to accept TCP and HTTP protocol. For examples of both public and private MSMQ Format Names, see **"MSMQ Format Name and Host Name" on page 25**. ***Note:*** If the *ComputerName* or *Host* name contains more than 15 characters, MSMQ will truncate the name. In this case, you must use the truncated ComputerName. refer to the queue Properties, General tab to see the specific ComputerName for your system. See **"MSMQ Format Name and Host Name" on page 25** for more information. See **"Identifying an MSMQ Queue" on page 22** for more information.
MSMQ Queue Name	Specifies the Microsoft Message Queue Name.	The string name that identifies the queue. This is associated with the host name, so the value is entered as **queuename**, where *queuename* is the queue name. See **"Identifying an MSMQ Queue" on page 22** for more information.
MSMQ Share Mode	Specifies the MSMQ share mode (**DENY_NONE** or **DENY_RECEIVE_SHARE**).	Select **DENY_NONE** or **DENY_RECEIVE_SHARE**. **DENY_NONE** is the configured default.

Table 3 Connectivity Map - Inbound - MSMQ Configuration (Continued)

Name	Description	Required Value
MSMQ Access Mode	Specifies the MSMQ Access Mode. Only RECEIVE_ACCESS is supported for inbound mode.	RECEIVE_ACCESS
MSMQ Receive Interval	Specifies the polling interval or frequency at which the eWay checks the queue for incoming messages (in milliseconds).	A number indicating the polling interval in milliseconds. The configured default is **5000** (5 seconds).
MSMQ Receive Action Code	Specifies the MSMQ Receive Action code. Only ACTION_RECEIVE is supported for inbound mode.	ACTION_RECEIVE

3.3.1 MSMQ Format Name and Host Name

The format name is a string that uniquely identifies a queue using connection details and the queue's path. Different types of format names can be used to specify how messages are routed, the type of destination, and the type of operation for which the queue is being opened.

The public or private format name property value is entered in the following manner:

For **public** queues use the following:

- DIRECT=TCP:*IPAddress**QueueName*
  for example: DIRECT=TCP:192.168.100.100\testmsmq
- DIRECT=OS:*ComputerName**QueueName*
  for example: DIRECT=OS:mypc-gx600\testmsmq
- DIRECT=HTTP://*Host*/msmq/*QueueName*
  for example: DIRECT=HTTP://mypc-gx600/msmq/testmsmq

For **private** queues use the following:

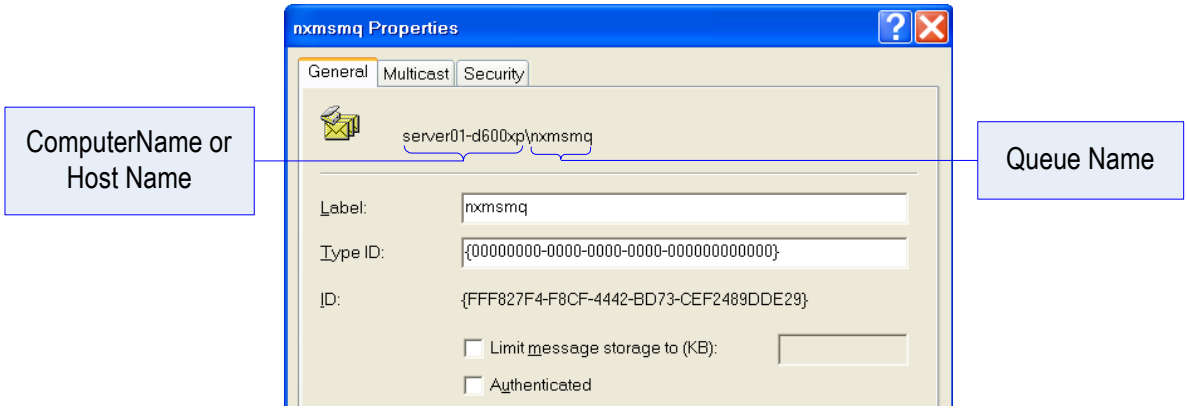
- DIRECT=TCP:*IPAddress*\private$*QueueName*
  for example: DIRECT=TCP:192.168.100.100\private$\privateqnxa
- DIRECT=OS:*ComputerName*\private$*QueueName*
  for example: DIRECT=OS:mypc-gx600\private$\privateqnxa

Configure your MQ security attributes to accept TCP and HTTP protocol. If you specify an **MSMQ Format Name**, the **MSMQ Queue Name** and **MSMQ Queue Alias** properties must be left blank.

To find the *ComputerName*, right-click **My Computer** and click **Manage**. From the **Computer Management** dialog box, select **Computer Management > Services and Applications > Message Queueing > Public Queues**. Right-click your queue and select **Properties**. From the General tab of the Properties dialog box, you can see the computer name and queue name used by MSMQ (see Figure 5).

Note: *The ComputerName and Host Name is limited to 15 characters by MSMQ. If the Host Name contains more than 15 characters, MSMQ will truncate the name. In this case, you must use the truncated Host Name. Refer to the queue Properties, General tab to see the specific Host Name for your system (see Figure 5).*

Figure 5 MSMQ Queue Properties



3.4 Outbound MSMQ eWay Connectivity Map Properties

The **MSMQ Configuration** section of the outbound **MSMQ eWay Connectivity Map properties** contains the top-level properties displayed in Table 4.

Identifying an MSMQ Queue

The MSMQ eWay identifies an MSMQ Queue using the first available value of these three properties, in the following order:

1. MSMQ Queue Alias
2. MSMQ Format Name
3. MSMQ Queue Name

When use one of these properties, for example **Format Name**, to identify a queue, we recommend that you leave the value for the other two properties blank (in this case, you would leave the **MSMQ Queue Alias** and **MSMQ Queue Name** properties blank). This ensures that only the queue identification method you specify is used.

Table 4 Connectivity Map - Inbound - MSMQ Configuration

Name	Description	Required Value
MSMQ Queue Alias	Specifies the queue alias. A queue alias associates an ADs path and a user-defined string (alias) with a public, private, or direct single-element format name. To send or receive messages using a queue alias, **MSMQ** and the **Active Directory** service must be installed on the same computer as the Logical Host.	The queue alias. See **"Identifying an MSMQ Queue" on page 27** for more information.

Table 4 Connectivity Map - Inbound - MSMQ Configuration (Continued)

Name	Description	Required Value
MSMQ Format Name	Specifies the format name of the queue. The format name is a string that uniquely identifies a queue using connection details and the queue's path. Different types of format names can be used to specify how messages are routed, the type of destination, and the type of operation for which the queue is being opened. Configure your MQ security attributes to accept TCP and HTTP protocol.	The public or private format name in the following manner: ▪ For **public** queues use the following: ♦ DIRECT=TCP:*IPAddress**QueueName* ♦ DIRECT=OS:*ComputerName**QueueName* ♦ DIRECT=HTTP://*Host*/msmq/*QueueName* ▪ For **private** queues use the following: ♦ DIRECT=TCP:*IPAddress*\private$*QueueName* ♦ *DIRECT=OS:ComputerName*\private$*Queue Name* Configure your MQ security attributes to accept TCP and HTTP protocol. For examples of both public and private MSMQ Format Names, see **"MSMQ Format Name and Host Name" on page 25**. If the *ComputerName* or *Host* name contains more than 15 characters, MSMQ will truncate the name. In this case, you must use the truncated ComputerName. refer to the queue Properties, General tab to see the specific ComputerName for your system. See **"MSMQ Format Name and Host Name" on page 25** for more information. See **"Identifying an MSMQ Queue" on page 27** for more information.
MSMQ Queue Name	Specifies the Microsoft Message queue name.	The string name that identifies the queue. This is associated with the host name, so the value is entered as **host/queue**, where *host* is the host name and *queue* is the queue name. See **"Identifying an MSMQ Queue" on page 27** for more information.

Table 4 Connectivity Map - Inbound - MSMQ Configuration (Continued)

Name	Description	Required Value
MSMQ Transaction Type	Specifies the outbound transaction type. The options are: ▪ **MQ_NO_TRANSACTION:** Select this value when sending messages to a non-transactional queue. ▪ **MQ_SINGLE_MESSAGE:** Select this value when sending messages to a transactional queue. This value does not apply to or affect inbound messages from a transactional queue.	Select **MQ_NO_TRANSACTION** or **MQ_SINGLE_MESSAGE** as the transaction type. Use the default value, **MQ_NO_TRANSACTION**. For more information refer to your MSMQ user documentation.
MSMQ Share Mode	Specifies the MSMQ share mode as one of the following: ▪ **DENY_NONE**: Grants everyone full access to send, peek at, or retrieve messages from the queue while it is open. ▪ **DENY_RECEIVE_SHARE**: You alone can peek at or retrieve messages until the queue is closed. Other applications are still able to send messages to the queue, but they may not delete messages while you have the queue open.	Select **DENY_NONE** or **DENY_RECEIVE_SHARE**. **DENY_NONE** is the configured default. **Error Conditions** An error message occurs if you or anyone tries to open the queue with **RECEIVE_ACCESS** or **PEEK_ACCESS** after Message Queuing opens the queue. If you attempt to open a queue with **DENY_RECEIVE_SHARE** when the queue is already open with **RECEIVE_ACCESS** or **PEEK_ACCESS**, the call will fail.
Message Priority	Specifies the message priority level for all messages. Priorities range from level **0** (low priority) to **7** (high priority). Messages are stored in descending order in the queue, with higher priority messages first. Priority is only valid when **MSMQ Access Mode** is set to **SEND_ACCESS** (see **"MSMQ Access Mode" on page 30**).	A number between **0** and **7** indicating the message priority. The configured default is **3**.

Table 4 Connectivity Map - Inbound - MSMQ Configuration (Continued)

Name	Description	Required Value
MSMQ Receive Action Code	Specifies the MSMQ receive action code as one of the following: **ACTION_RECEIVE**: Reads the message at the current cursor location and removes it from the queue. **ACTION_PEEK_CURRENT**: "Peeks" for last message in the queue. You can use this functionality when there are several messages coming in and going out of queue, and you want to peek (look) and get the latest message.	Select **ACTION_RECEIVE** or **ACTION_PEEK_CURRENT**. **ACTION_RECEIVE** is the configured default.
MSMQ Access Mode	Specifies whether Message Queuing opens the queue with peek, send, or receive access. ▪ **RECEIVE_ACCESS**: Allows messages to be retrieved from a queue when they are read. This is used when the receiving application opens a queue to remove messages. ▪ **SEND_ACCESS**: Allows messages to be sent to a queue. This is used when the sending application opens a queue to send messages. ▪ **PEEK_ACCESS**: Allows messages to be read from a queue without removing them. This is used when the receiving application opens a queue to read message.	Select **RECEIVE_ACCESS**, **SEND_ACCESS**, or **PEEK_ACCESS**. You must set Access Mode to **SEND ACCESS** to use Message Priority.

Table 4 Connectivity Map - Inbound - MSMQ Configuration (Continued)

Name	Description	Required Value
Connection Mode	Specifies whether a physical connection is established when an external connection is instantiated. The options are: ▪ **Automatic**: Establishes a physical connection when an external connection is instantiated. ▪ **Manual**: Does not connect to the external system on startup, and instead expects the user to initiate the connection from the Collaboration (for example, by calling the **connect()** method).	Select **Automatic** or **Manual** (dynamic). The configured default is **Automatic**.

3.5 MSMQ eWay Environment Properties

The MSMQ eWay Environment properties are organized into the following sections:

- **Inbound MSMQ eWay > MSMQ Environment** on page 32
- **Inbound MSMQ eWay > MDB Settings** on page 33
- **Inbound MSMQ eWay > Connection Retry Settings** on page 34
- **Outbound MSMQ eWay > MSMQ Environment** on page 35
- **Outbound MSMQ eWay > Connection Retry Settings** on page 36

3.5.1 Inbound MSMQ eWay > MSMQ Environment

The **Inbound MSMQ eWay > MSMQ Environment** section of the **MSMQ eWay Environment properties** contains the top-level properties displayed in Table 5.

Table 5 Environment - Inbound MSMQ eWay > MSMQ Environment

Name	Description	Required Value
MSMQ Host Name	Specifies the Microsoft Message Queue host name.	The Microsoft Message Queue host name. Avoid using an IP address or "localhost" for the local server. These may not work in your MSMQ Environment. ***Note:*** If the Host Name contains more than 15 characters, MSMQ will truncate the name. In this case, you must use the truncated Host Name. Refer to the queue Properties, General tab to see the specific Host Name for your system. See **"MSMQ Format Name and Host Name" on page 25** for more information.

3.5.2 Inbound MSMQ eWay > MDB Settings

The **Inbound MSMQ eWay > MDB Settings** section of the **MSMQ eWay Environment properties** contains the top-level properties displayed in Table 6.

Table 6 Environment - Inbound MSMQ eWay > MDB Settings

Name	Description	Required Value
Max Pool Size	Specifies the maximum number of physical connections the pool can contain. A value of **0** (zero) indicates that there is no maximum.	An integer indicating the maximum pool size. The configured default is **1000**.

3.5.3 Inbound MSMQ eWay > Connection Retry Settings

The **Inbound MSMQ eWay > Connection Retry Settings** section of the **MSMQ eWay Environment properties** contains the top-level properties displayed in Table 7.

Table 7 Environment - Inbound MSMQ eWay > Connection Retry Settings

Name	Description	Required Value
Connection Retry Count	Specifies the maximum number of attempts made to connect to the destination queue manager or queue. If the queue manager or queue cannot be accessed for any reason, this setting specifies how many reattempts are made to complete the processing.	An integer indicating the maximum number of connection attempts. The configured default is **0**.
Connection Retry Interval	Specifies the amount of time (in milliseconds) between attempts to connect to the destination queue manager or queue. This is the pause between each reattempt to access the destination queue manager or queue. Used in conjunction with the 'Connection Retry Count' setting.	An integer indicating the wait time in milliseconds between connection attempts. The configured default is **1000**.

3.5.4 Outbound MSMQ eWay > MSMQ Environment

The **Outbound MSMQ eWay > MSMQ Environment** section of the **MSMQ eWay Environment properties** contains the top-level properties displayed in Table 8.

Table 8 Environment - Outbound MSMQ eWay > MSMQ Environment

Name	Description	Required Value
MSMQ Host Name	Specifies the Microsoft Message Queue host name.	The Microsoft Message Queue host name. Avoid using an IP address or "localhost" for the local server. These may not work in your MSMQ Environment. ***Note:*** If the Host Name contains more than 15 characters, MSMQ will truncate the name. In this case, you must use the truncated Host Name. Refer to the queue Properties, General tab to see the specific Host Name for your system. See **"MSMQ Format Name and Host Name" on page 25** for more information.

3.5.5 Outbound MSMQ eWay > Connection Retry Settings

The **Outbound MSMQ eWay > Connection Retry Settings** section of the **MSMQ eWay Environment properties** contains the top-level properties displayed in Table 9.

Table 9 Environment - Outbound MSMQ eWay > Connection Retry Settings

Name	Description	Required Value
Connection Retry Count	Specifies the maximum number of attempts made to connect to the destination queue manager or queue. If the queue manager or queue cannot be accessed for any reason, this setting specifies how many reattempts are made to complete the processing.	An integer indicating the maximum number of connection attempts. The configured default is **0**.
Connection Retry Interval	Specifies the amount of time (in milliseconds) between attempts to connect to the destination queue manager or queue. This is the pause between each reattempt to access the destination queue manager or queue. Used in conjunction with the 'Connection Retry Count' setting.	An integer indicating the wait time in milliseconds between connection attempts. The configured default is **1000**.

3.6 Enabling Rollback When an MSMQ Message Fails

In order to roll back an outbound MSMQ message when a failure occurs in the Java Collaboration Definition (for example, failure to insert a duplicate row into a database table that is defined to have unique keys), do the following:

1. Select **Manual** as the MSMQ Connection Mode in the outbound MSMQ eWay Connectivity Map properties (see **"Connection Mode" on page 31**).
2. Use the following JCD code:

```
try {
    MSMQClient_1.getEwayConfiguration().setOutMSMQName( "public"
);
    MSMQClient_1.getEwayConfiguration().setOutMSMQShareMode(
"DENY_RECEIVE_SHARE" );

MSMQClient_1.getEwayConfiguration().setOutMSMQReceiveActionCode(
"ACTION_PEEK_CURRENT" );
    MSMQClient_1.connect();
    MSMQClient_1.getMSMQMessage();
    TestDB_1.getTEST1().insert();
    TestDB_1.getTEST1().setTESTSTRING( "From JCD" );
    TestDB_1.getTEST1().insertRow();

MSMQClient_1.getEwayConfiguration().setOutMSMQReceiveActionCode(
"ACTION_RECEIVE" );
    MSMQClient_1.getMSMQMessage();
    MSMQClient_1.disconnect();
} catch ( Exception Catch ) {
    MSMQClient_1.disconnect();
}
```

   The key code items are highlighted in **bold**. You must also use a **try/catch block** to catch the exception (for this example, the error is a failed insert to database).
3. Open the queue with the following settings:
   - MSMQ Share Mode – **DENY_RECEIVE_SHARE**: This locks the queue so that other applications cannot open the queue in Receive mode. Until your application closes the queue, no other application can open the queue to receive the message.
   - MSMQ Receive Action Code – **ACTION_PEEK_CURRENT**: This opens the queue so you can "peek" (view) the message.
4. Next, try a database update on the message you received from the queue. If this fails, move on to step **7** (close the queue instance). If that succeeds, change the eWay configuration properties to receive messages programmatically in this manner (see **"MSMQ Receive Action Code" on page 24**).
5. Next, receive the message. This removes the message from queue.
6. Disconnect from the queue.
7. Finally, close the queue using **MSMQClient_1.disconnect()**. This closes the current queue instance and unlocks the queue.

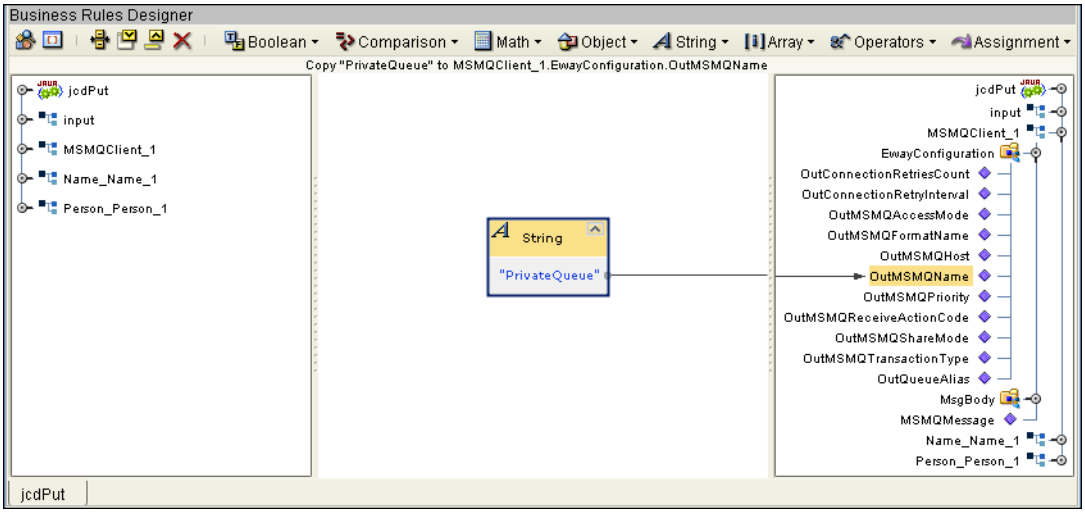
3.7 Manually Configuring the eWay

Manual mode allows you to override eWay configurations in the Java Collaboration Definition. To enable this, set the **MSMQ Connection Mode** (in the outbound Connectivity Map properties) to "**Manual**". The MSMQ Connection Mode is only available for outbound eWays (see **"Connection Mode" on page 31**).

Collaborations cannot include both **Manual** mode and **Automatic** mode. Also, Collaborations using **Manual** mode may not participate in connection pooling.

In Manual mode, all available properties and methods can be changed through the Collaboration's OTDs. For example, to set the **OutMSMQName**, you assign a **String Literal** to the **OutMSMQName** parameter on the right hand side of the Collaboration in the Collaboration Editor (see Figure 6). You can also change the property value using other sources, such as the content in an input file or the value from another OTD.

Figure 6 Manually Configuring an eWay



Once all of the new property values are assigned, you must use the **Connect()** method to establish a connection to the queue. For example, from the Business Rules Designer pane, right-click the **MSMQClient_1** OTD and select Select method to call from the shortcut menu. Select **connect()** from the method selection box.

When you use the **getMSMQMessage()** or **putMSMQMessage()** methods to retrieve or send a message, you must use the **disconnect()** method to disconnect from the queue.

Chapter 4

Implementing an MSMQ eWay Project using eInsight

This chapter describes how to use the MSMQ eWay with the eInsight Business Process Manager and eInsight.

Note: *You must have the* **eInsight.sar** *file installed to use the Web Services interface.*

What's in This Chapter

- **eInsight Engine and Components** on page 39
- **The MSMQ eWay With eInsight** on page 40
- **MSMQ eWay Sample Project for eInsight** on page 40
- **Importing a Sample Project** on page 41
- **Creating the prjMSMQBPM Project** on page 41

4.1 eInsight Engine and Components

eGate components can be deployed as Activities in eInsight Business Processes. Once a component is associated with an Activity, eInsight invokes it using a Web Services interface. eGate components that can interface with eInsight in this way include the following:

- Java Messaging Service (JMS)
- Object Type Definitions (OTDs)
- eWays
- Collaborations

Using the Enterprise Designer and eInsight, you can add an Activity to a Business Process, then associate that Activity with an eGate component, for example, an eWay. Then, when eInsight runs the Business Process, it automatically invokes that component via its Web Services interface.

See the *eInsight Business Process Manager User's Guide* for details.

4.2 The MSMQ eWay With eInsight

An eInsight Business Process Activity can be associated with the MSMQ eWay during the system design phase. To make this association, select the desired operators under the eWay in the Enterprise Explorer and drag it onto the eInsight Business Process Designer canvas.

The MSMQ eWay has the following operators available:

- receiveMSMQMessage
- putWSMSMQMessage
- getWSMSMQMessage

The operation is automatically changed to an Activity with an icon identifying the component that is the basis for the Activity. At run time, eInsight invokes each step in the order defined in the Business Process. Using eInsight's Web Services interface, the Activity, in turn, invokes the MSMQ eWay.

4.3 MSMQ eWay Sample Project for eInsight

The MSMQ eWay includes one eInsight sample Project, **prjMSMQBPM**. This sample Project performs three separate operations that demonstrate the following:

- Get Operation: The eWay subscribes to an MSMQ queue. The File eWay is used as a trigger. When a "trigger" message is received by the File eWay, the MSMQ eWay picks up any available XML data (messages) from the queue and publishes them to an eInsight Business Process. The Business Process unmarshals the message, modifies the data by converting a name from one value (first, last) to two values (first and last), and marshals the data. The message is then published to a outbound File eWay. The File eWay publishes the message an external directory.
- Put Operation: A File eWay subscribes to an external directory. XML data (message) is picked up from the input directory and published to an eInsight Business Process. The Business Process unmarshals the message, modifies the data converting a name from two values (first and last) to one value (first, last), and marshals the data. The message is then published to an MSMQ queue.
- **Receive Operation:** The eWay subscribes to an MSMQ queue. XML data (message) is picked up from the queue and published to an eInsight Business Process. The Business Process unmarshals the message, modifies the data converting a name from one value (first, last) to two values (first and last), and marshals the data. The message is then published to a outbound File eWay. The File eWay publishes the message to a directory.

Both the **Get** and **Receive** operations read the message from the queue. The **Get** operation requires a trigger, such as the presence of an input file. **Receive**, on the other hand, routinely polls the queue for messages. The polling rate is specified by the inbound property, MSMQ Receive Interval. No trigger is necessary.

This chapter provides directions for importing and completing the prjMSMQBPM Project. It also explains how to create the sample Project manually.

4.4 Importing a Sample Project

Sample eWay Projects are included as part of the installation package. To import a sample eWay Project to the Enterprise Designer do the following:

1. The sample files are uploaded with the eWay's documentation SAR file and downloaded from the Sun Java Composite Application Platform Suite Installer's **Documentation** tab. The **MSMQ_eWay_Sample.zip** file contains the various sample Project ZIP files. Extract the samples to a local file.
2. Save all unsaved work before importing a Project.
3. From the Enterprise Designer's Project Explorer pane, right-click the Repository and select **Import** from the shortcut menu. The **Import Manager** appears.
4. Browse to the directory that contains the sample Project zip file. Select the sample file (for this sample, **prjMSMQBPM.zip**) and click **Import**. After the sample Project is successfully imported, click **Close**.
5. Before an imported sample Project can be run you must do the following:
   - Create an **Environment** (see **"Creating an Environment" on page 56**)
   - Configure the eWays for your specific system (see **"Configuring the eWays" on page 57**)
   - Create a **Deployment Profile** (see **"Creating and Activating the Deployment Profile" on page 58**)
   - Create and start a domain (see **"Creating and Starting the Domain" on page 59**)
   - Build and deploy the Project (see **"Building and Deploying the Project" on page 60**)

4.5 Creating the prjMSMQBPM Project

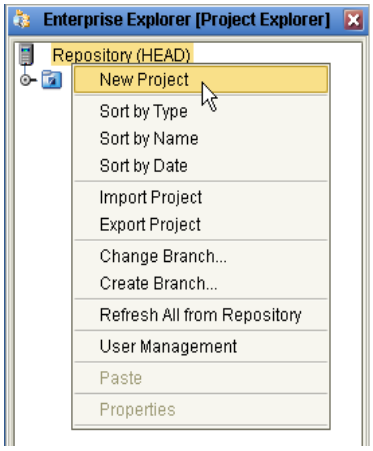
The following section explains how the sample Project's components are created manually. The first step is to create and name a new Project in eGate Enterprise Designer.

4.5.1 Creating a Project

First, create a new Project in the SeeBeyond Enterprise Designer.

1. From the Enterprise Explorer's Project Explorer tab, right-click the Repository and select **New Project** (see **Figure 7 on page 42**).

Figure 7 Enterprise Explorer - New Project



2 A new Project (Project1) appears on the Project Explorer tree. Rename the Project to **prjMSMQBPM**.

4.5.2 Creating OTDs Using the DTD Wizard

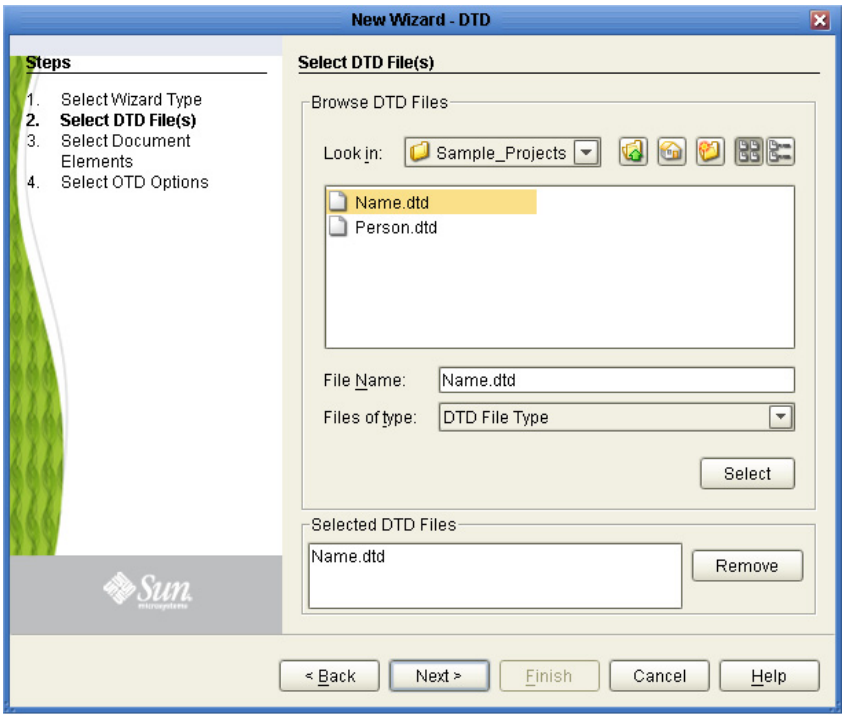
Two sample XML Document Type Definitions (DTDs) are included with the sample Projects: **Name.dtd** and **Person.dtd**. Convert these to Object Type Definitions (OTDs) using the **Object Type Definition Wizard**.

1 From the Project Explorer tree, right-click the **prjMSMQBPM** Project and select **New** > **Object Type Definition** from the shortcut menu. The **Object Type Definition Wizard** appears.

2 From the Select Wizard Type box, select **DTD**. The **DTD Wizard** appears.

3 For Step 2 of the wizard, "Select DTD File(s)," browse to the location where you saved the sample Project data, and select the Name.dtd file (see **Figure 8 on page 43**) and click **Select**. Click **Next**.

Figure 8 DTD Wizard - Select DTD File(s)



4 For step 3 of the wizard, "Select Document Elements," click **Next**.

5 For step 4 of the wizard, "Select OTD Options," use the default settings and click **Finish**. The OTD Editor appears displaying the new **Name_Name.otd.**

6 Repeat steps 1 - 4 using **Person.dtd** to create the Person_Person.otd.

7 Save your work to the Repository and close the OTD Editor.

4.5.3 Creating the Business Processes

The next step to creating the sample is to create three Business Processes using the **eInsight Business Process Designer**. These are:

- **bpGet**: for the Get operation
- **bpPut**: for the Put operation
- **bpReceive**: for the Receive operation

Creating the bpGet Business Process

The Business Process's Business Rules are written using the Business Rules Designer.

Creating the Business Process Flow

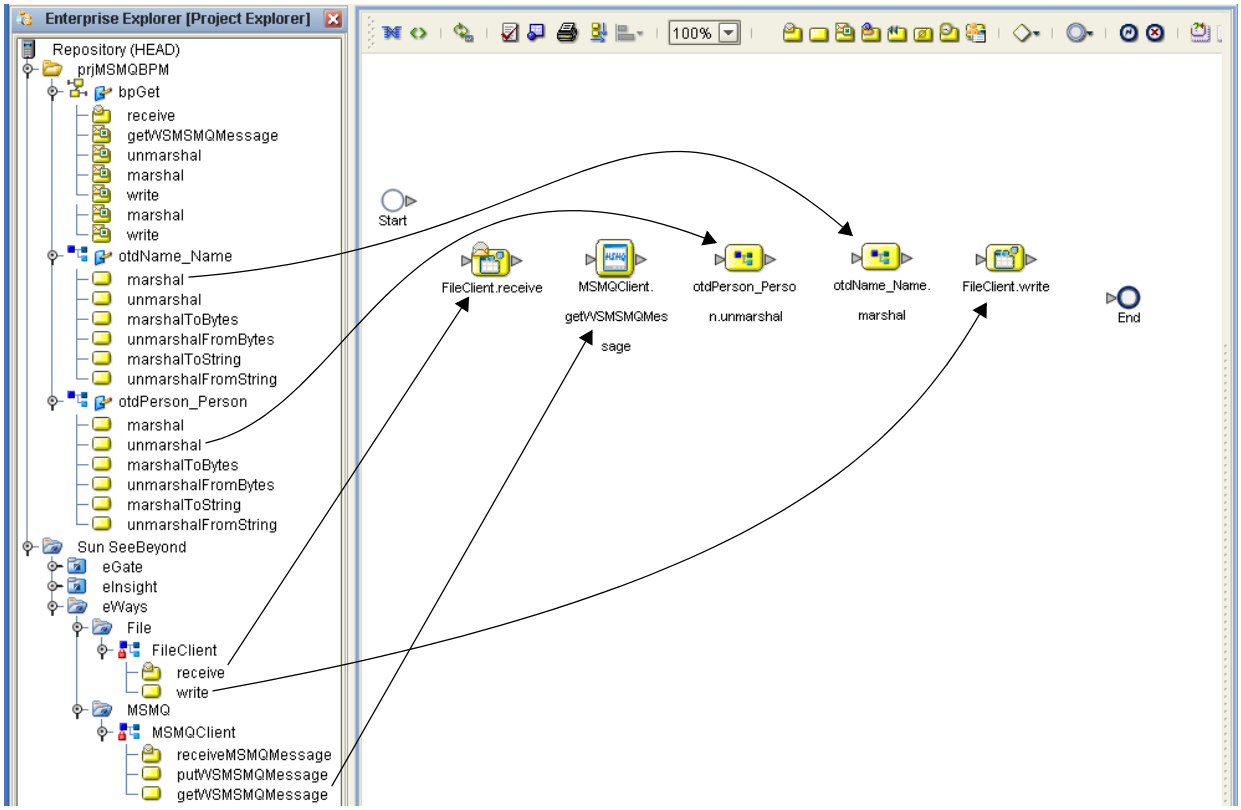
1 From the Project Explorer tree, right-click **prjMSMQBPM**, and select **New** > **Business Process** from the shortcut menu. The eInsight Business Process Designer appears and **BusinessProcess1** is added to the Project Explorer tree. Rename the Business Process to **bpGet**.

2 From the Project Explorer tree, expand **SeeBeyond > eWays > MSMQ > MSMQClient** nodes. In the same way, expand **SeeBeyond > eWays > File > FileClient** nodes to expose the available Business Process elements.

3 Populate the eInsight Business Process Designer **bpGet** canvas with the following elements from the Project Explorer tree, as displayed in **Figure 9 on page 44**:

- receive, under SeeBeyond > eWays > File > FileClient
- getWMSMQMessage under SeeBeyond > eWays > MSMQ > MSMQClient
- unmarshal under Person_Person
- marshal under Name_Name
- write, under SeeBeyond > eWays > File > FileClient

Figure 9 eInsight Business Process Designer - Populate the Canvas

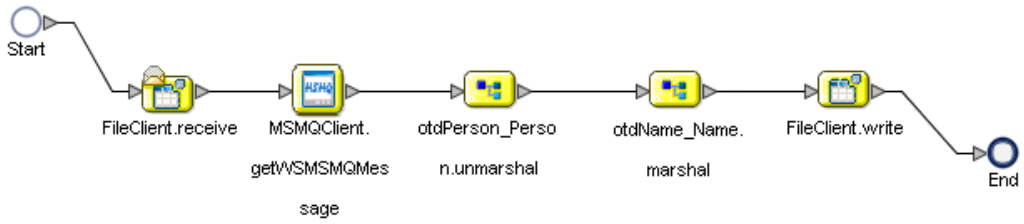


4 Link the modeling elements by clicking on the element's connector, and dragging the cursor to the next element's connector, making the following links as displayed in Figure 10. Link the following:

- **Start** to **FileClient.receive**
- **FileClient.receive** to **MSMQClient.getWSMSMQMessage**
- **MSMQClient.getWSMSMQMessage** to **otdPerson_Person.unmarshal**
- **otdPerson_Person.unmarshal** to **otdName_Name.marshal**
- **otdName_Name.marshal** to **FileClient.write**

- **FileClient.write** to **End**

Figure 10 eInsight Business Process Designer - bpGet Business Process



5 Create the **bpPut** and **bpReceive** Business Processes in the same manner, adding components and names to match those in **Figure 11 on page 45** for the **bpPut** Business Process, and in **Figure 12 on page 45** for the **bpReceive** Business Process.

Figure 11 eInsight Business Process Designer - bpPut Business Process

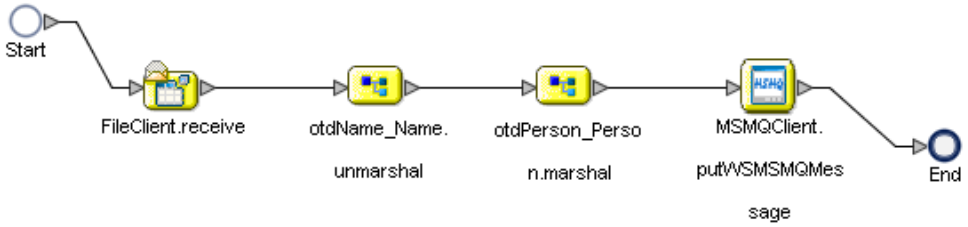
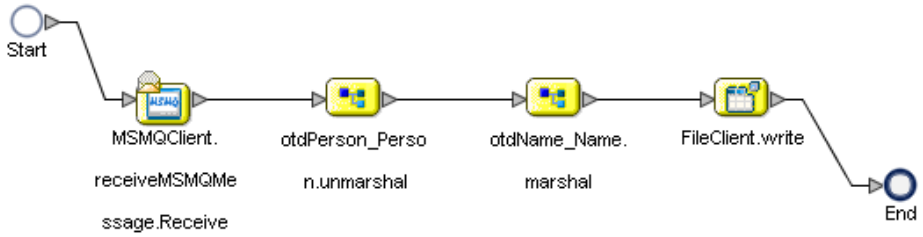


Figure 12 eInsight Business Process Designer - bpReceive Business Process



6 Save your current changes to the Repository.

4.5.4 Creating the Business Rules Using the Business Rules Designer

Business Rules that you create between the Business Process elements define the relationship between the input and output Attributes of the elements.

Creating the bpGet Business Process Business Rules

1 From the Project Explorer tree, double-click the **bpGet** Business Process. The Business Process Designer opens, displaying the **bpGet** Business Process.

2 Create the **MSMQClient.Receive** to **otdPerson_Person.unmarshal** Business Rule:

A Right-click the link between the **MSMQClient.Receive** and **otdPerson_Person.unmarshal** activities, and select **Add Business Rule** from the shortcut menu as displayed in Figure 13.

Figure 13 eInsight Business Process Designer - Adding Business Rules

B Double-click the Business Rule icon. The Business Rule Designer appears at the bottom of the eInsight Business Process Designer.

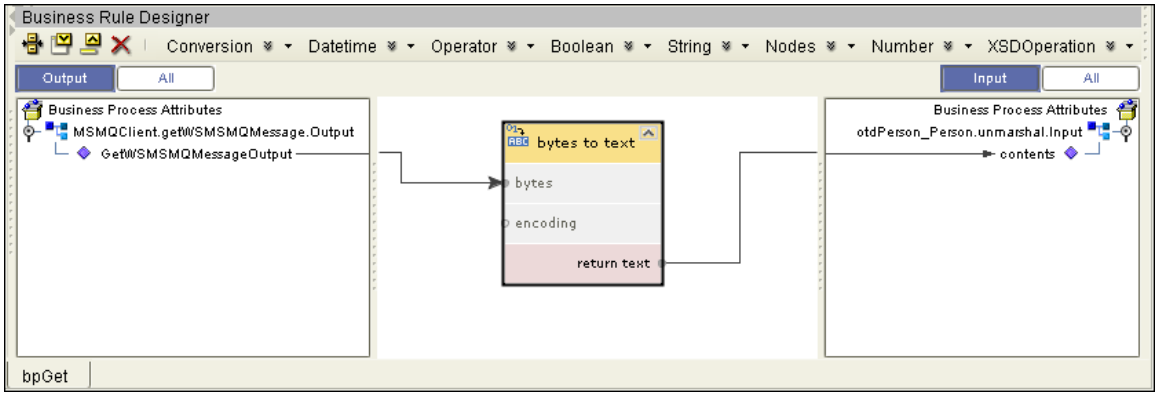
C From the Business Rule Designer's String menu, select **Settings**. The Method Palette appears. From the **String** tab of the Method Palette, select **bytes to text** (click the check box for this option) and click **Close**. The **bytes to text** option is added to the **String** menu.

D From the Business Rule Designer's String menu, select **bytes to text**. The **bytes to text** method box appears.

E From the Output pane of the Business Rules Designer, map **GetWSMSMQMessageOutput,** under **MSMQClient.getWSMSMQMessage.Output**, to the **bytes** input node of the **bytes to text** method box. To do this, click on **ByteArrayData** and drag your cursor to the **bytes** input node of the **bytes to text** method box.

F Map the **return text** output node of the **bytes to text** method box to **contents**, under **otdPerson_Person.unmarshal.input** in the Input pane of the Business Rule Designer (see Figure 14).

Figure 14 eInsight Business Rule Designer



3 Create the **otdPerson_Person.unmarshal** to **otdName_Name.marshal** Business Rule:

A Right-click the link between the **otdPerson_Person.unmarshal** and **otdName_Name.marshal** activities and select **Add Business Rule** from the shortcut menu.

B From the Business Rule Designer toolbar's **String** menu, select **Settings**. The **Method Palette** appears. From the **String** tab of the Method Palette, select **substring before** and **substring after**, and click **Close**. The **substring before** and **substring after** options are added to the **String** menu.

C From the Business Rule Designer's **String** menu, select **substring before**. The **substring before** method box appears.

D Map **Name** under **otdPerson_Person.unmarshal.Output > Person**, in the Output pane of the Business Rule Designer, to the **string1** input node of the **substring before** method box.

E Double-click the **string 2** field of the **substring before** method box and enter one space (press the space bar one time), as the String value.

F Map the **return string** output node of the **substring before** method box, to **FirstName** under **otdName_Name.marshal.input > Name**, in the Input pane of the Business Rule Designer.

G From the Business Rule Designer's **String** menu, select **substring after**. The **substring after** method box appears.

H Map **Name** under **otdPerson_Person.unmarshal.Output > Person** in the Output pane of the Business Rule Designer, to the **string1** input node of the **substring after** method box.

I Double-click the **string 2** field of the **substring before** method box and enter one space (press the space bar one time), as the String value.

J Map the **return string** output node of the **substring after** method box, to **LastName** under **otdName_Name.marshal.input > Name**, in the Input pane of the Business Rule Designer.

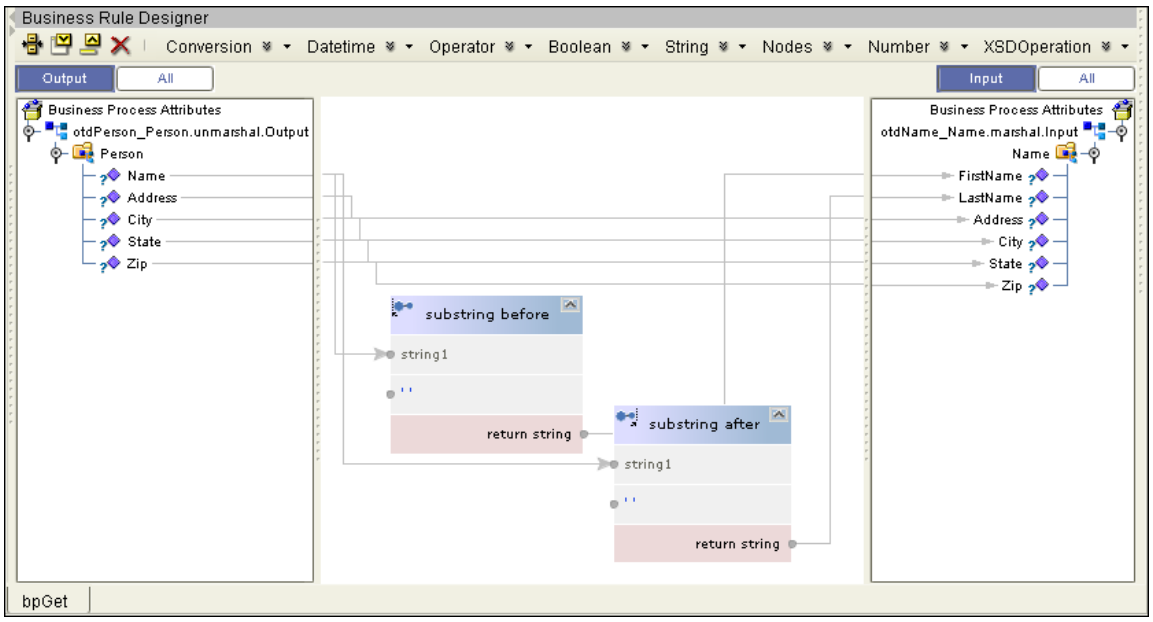
K Map **Address** under otdPerson_Person.unmarshal**.Output > Person**, in the Output pane of the Business Rule Designer, to **Address** under **otdName_Name.marshal.input > Name**, in the Input pane of the Business Rule Designer.

L Map **City** under **otdPerson_Person.unmarshal.Output > Person**, in the Output pane, to **City** under **otdName_Name.marshal.input > Name**, in the Input pane.

M Map **State** under **otdPerson_Person.unmarshal.Output > Person**, in the Output pane, to **State** under **otdName_Name.marshal.input > Name**, in the Input pane.

N Map **Zip** under **otdPerson_Person.unmarshal.Output > Person**, in the Output pane, to **Zip** under **otdName_Name.marshal.input > Name**, in the Input pane (see **Figure 15 on page 48**).

Figure 15 eInsight Business Rule Designer



4 Create the **otdName_Name.marshal** to **FileClient.write** Business Rule:

A Right-click the link between the **otdName_Name.marshal** and **FileClient.write** activities and select **Add Business Rule** from the shortcut menu.

B From the Output pane of the Business Rules Designer, map **contents** under **otdName_Name.marshal.output**, to **text** under **FileClient.write.Input**, in the Input pane of the Business Rule Designer (see Figure 16).

Figure 16 eInsight Business Rule Designer

5 Once all the Business Rules in the Business Process are complete, click the **Synchronize Graphical Model and Business Process Code** icon, on the Business Process Designer toolbar, to synchronize the graphical interface to the Business Process code.

6 Click the Enterprise Designer's **Save All** icon to save your current changes.

Creating the bpPut Business Process Business Rules

1 Open the **bpPut** Business Process in the Business Process Designer.

2 Create the **FileClient.receive** to **otdName_Name.unmarshal** Business Rule:

A Right-click the link between the **FileClient.receive** and **otdName_Name.unmarshal** activities and select **Add Business Rule** from the shortcut menu.

B From the Output pane of the Business Rules Designer, map **text** under **FileClient.write.output** in the Output pane of the Business Rule Designer, to **contents** under **otdName_Name.unmarshal.Input** in the Input pane of the Business Rule Designer.

3 Create the **otdName_Name.unmarshal** to **otdPerson_Person.marshal** Business Rule:

A Right-click the link between the **otdName_Name.unmarshal** and **otdPerson_Person.marshal** activities and select **Add Business Rule** from the shortcut menu.

B From the Business Rule Designer's **String** menu, select **concat**. The **concat** method box appears.

C From the Output pane of the Business Rules Designer, map **FirstName** under **otdName_Name.unmarshal.Output > Name**, to the **string** input node of the **concat** method box.

D Double-click the **str (string)** field of the **concat** method box and enter one space (press the space bar one time), as the String value.

E From the Output pane of the Business Rules Designer, map **LastName** under **otdName_Name.unmarshal.Output > Name**, to the **str (string)** input node of the **concat** method box.

F Map the **return string** node of the **concat** method box, to **Name,** under **otdPerson_Person.marshal.input > Person**, in the Input pane of the Business Rule Designer.

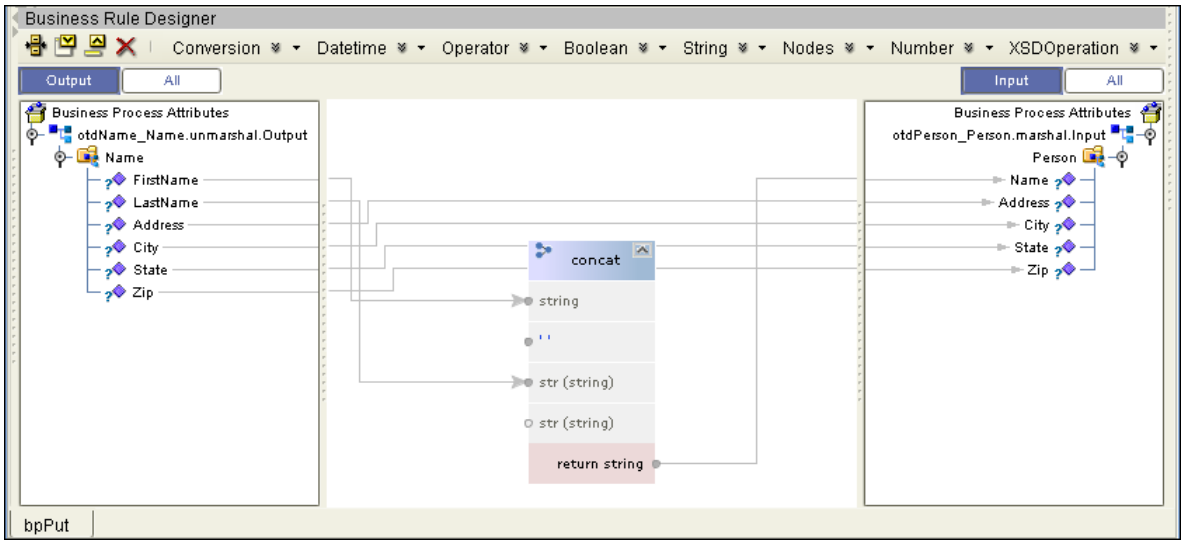
G From the Output pane of the Business Rules Designer, map **Address** under **otdName_Name.unmarshal.Output > Name**, to **Address,** under **otdPerson_Person.marshal.input > Person**, in the Input pane of the Business Rule Designer.

H From the Output pane of the Business Rules Designer, map **City,** under **otdName_Name.unmarshal.Output > Name**, to **City** under **otdPerson_Person.marshal.input > Person**, in the Input pane.

I From the Output pane of the Business Rules Designer, map **State,** under **otdName_Name.unmarshal.Output > Name**, to **State** under **otdPerson_Person.marshal.input > Person**, in the Input pane.

J From the Output pane of the Business Rules Designer, map **Zip,** under **otdName_Name.unmarshal.Output > Name**, to **Zip,** under **otdPerson_Person.marshal.input > Person**, in the Input pane (see **Figure 17 on page 50**.

Figure 17 eInsight Business Rule Designer



4 Create the **otdPerson_Person.marshal** to **MSMQClient.putWSMSMQMessage** Business Rule:

A Right-click the link between the **otdPerson_Person.marshal** and **MSMQClient.putWSMSMQMessage** activities, and select **Add Business Rule** from the shortcut menu.

B From the Business Rule Designer's **String** menu, select **Settings**. The **Method Palette** appears. From the **String** tab of the Method Palette, select **text to bytes** and click **Close**. The **text to bytes** option is added to the **String** menu.

C From the Business Rule Designer's **String** menu, select **text to bytes**. The **text to bytes** method box appears.

D From the Output pane of the Business Rules Designer, map **contents** under **otdPerson_Person.marshal.Output** to the **text** input node of the **text to bytes** method box.

E Map the **return bytes** output node of the **text to bytes** method box to **payload** under **MSMQClient.putWSMSMQMessage.Intput**, in the Input pane of the Business Rule Designer.

5 Once all the Business Rules in the Business Process are complete, click the **Synchronize Graphical Model and Business Process Code** icon on the Business Process Designer toolbar.

6 Save your current changes to the repository.

Creating the bpReceive Business Process Business Rules

1 Open **bpReceive** in the Business Process Designer.

2 Create the **MSMQClient.receiveMSMQMessage.Receive** to **otdPerson_Person.unmarshal** Business Rule:

A Right-click the link between the **MSMQClient.receiveMSMQMessage.Receive** and **otdPerson_Person.unmarshal** activities and select **Add Business Rule** from the shortcut menu.

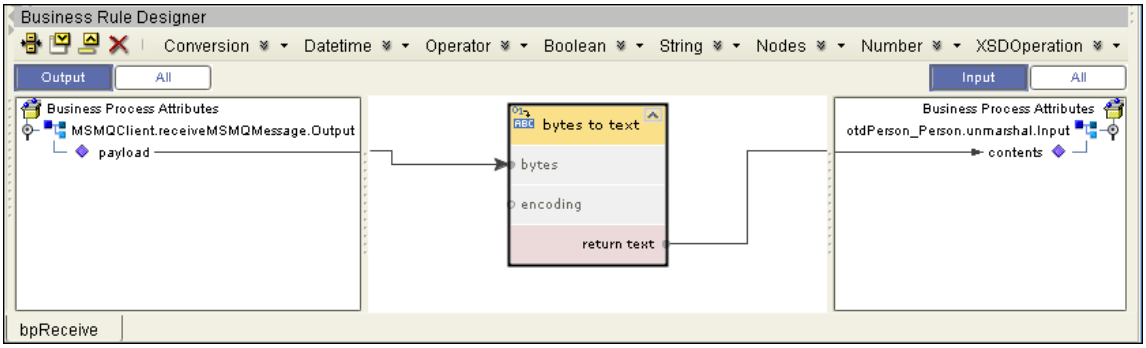
B From the Business Rule Designer's **String** menu, select **bytes to text**. The **bytes to text** method box appears.

If **bytes to text** is not an available option, click **Settings** from the String menu, and select **bytes to text** from the Methods Pallet. The **bytes to text** option is added to the String menu.

C From the Output pane of the Business Rules Designer, map **payload,** under **MSMQClient.receiveMSMQMessage.Output**, to the **bytes** input node of the **bytes to text** method box.

D Map the **return text** output node of the **bytes to text** method box to **contents** under **otdPerson_Person.unmarshal.Input** in the Input pane of the Business Rules Designer (see Figure 18).

Figure 18 eInsight Business Rule Designer



3 Create the **otdPerson_Person.unmarshal** to **otdName_Name.marshal** Business Rule:

A Right-click the link between the **otdPerson_Person.unmarshal** and **otdName_Name.marshal** activities and select **Add Business Rule** from the shortcut menu.

B From the Business Rule Designer's **String** menu, select **substring before**. The **substring before** method box appears.

C Map **Name** under **otdPerson_Person.unmarshal.Output > Person**, in the Output pane of the Business Rule Designer, to the **string1** input node of the **substring before** method box.

D Double-click the **string 2** field of the **substring before** method box and enter one space (press the space bar one time), as the String value.

E Map the **return string** output node of the **substring before** method box, to **FirstName** under **otdName_Name.marshal.input > Name**, in the Input pane of the Business Rule Designer.

F From the Business Rule Designer's **String** menu, select **substring after**. The **substring after** method box appears.

G Map **Name** under **otdPerson_Person.unmarshal.Output > Person** in the Output pane of the Business Rule Designer, to the **string1** input node of the **substring after** method box.

H Double-click the **string 2** field of the **substring before** method box and enter one space (press the space bar one time), as the String value.

I Map the **return string** output node of the **substring after** method box, to **LastName** under **otdName_Name.marshal.input > Name**, in the Input pane of the Business Rule Designer.

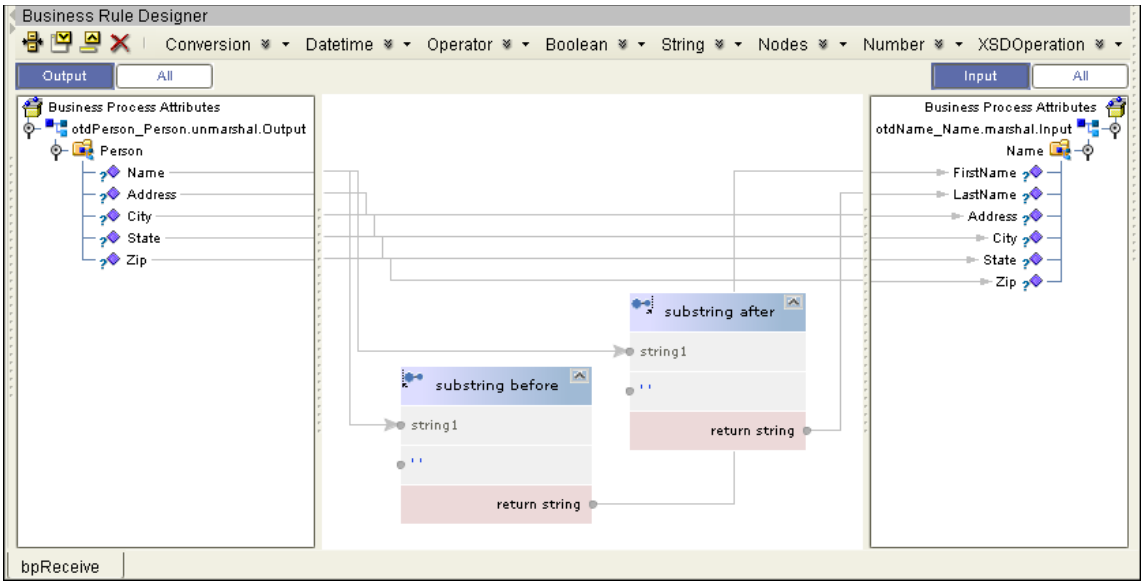
J Map **Address** under otdPerson_Person.unmarshal**.Output > Person**, in the Output pane of the Business Rule Designer, to **Address** under **otdName_Name.marshal.input > Name**, in the Input pane of the Business Rule Designer.

K Map **City** under **otdPerson_Person.unmarshal.Output > Person**, in the Output pane, to **City** under **otdName_Name.marshal.input > Name**, in the Input pane.

L Map **State** under **otdPerson_Person.unmarshal.Output > Person**, in the Output pane, to **State** under **otdName_Name.marshal.input > Name**, in the Input pane.

M Map **Zip** under **otdPerson_Person.unmarshal.Output > Person**, in the Output pane, to **Zip** under **otdName_Name.marshal.input > Name**, in the Input pane (see Figure 19).

Figure 19 eInsight Business Rule Designer



4 Create the **otdName_Name.marshal** to **FileClient.write** Business Rule:

A Right-click the link between the **otdName_Name.marshal** and **FileClient.write** activities and select **Add Business Rule** from the shortcut menu.

B From the Output pane of the Business Rules Designer, map **contents,** under **otdName_Name.marshal.output**, to **text,** under **FileClient.write.Input** in the Input pane of the Business Rule Designer.

5 Once you have completed all the Business Rules in the Business Process, click the **Synchronize Graphical Model and Business Process Code** icon on the Business Process Designer toolbar.

6 Save your current changes to the Repository.

4.5.5 Creating the Connectivity Maps

Connectivity Maps provide a canvas for assembling and configuring a Project's components. The prjMSMQBPM Project uses three Connectivity Maps:

- **cmGet**: for the Get operation
- **cmPut**: for the Put operation
- **cmReceive**: for the Receive operation

To create the **cmGet** Connectivity Map, do the following:

1 From the Project Explorer, right-click **prjMSMQBPM** and select **New > Connectivity Map** from the shortcut menu.

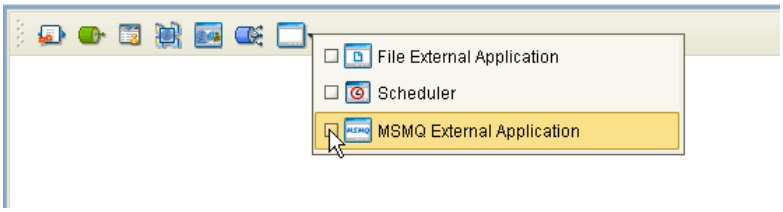
2 The New Connectivity Map appears and a node named **CMap1** is added to the Project on the Project Explorer tree. From the Project Explorer tree, rename the Connectivity Map to **cmGet**.

Select the External Applications

In a Connectivity Map, the icons on the toolbar represent available components used to populate the Connectivity Map canvas.

eWays in the Connectivity Map are associated with External Application. For example, to establish a connection to an MSMQ queue, you must first select MSMQ as an External Application to use in your Connectivity Map (see Figure 20).

Figure 20 Connectivity Map - External Applications



1 Click the **External Application** icon on the Connectivity Map toolbar.

2 Select the External Applications that you require for your Project (for this sample, the **MSMQ**, and **File** External Applications). Icons representing the selected External Applications are added to the Connectivity Map toolbar.

Populate the Connectivity Map

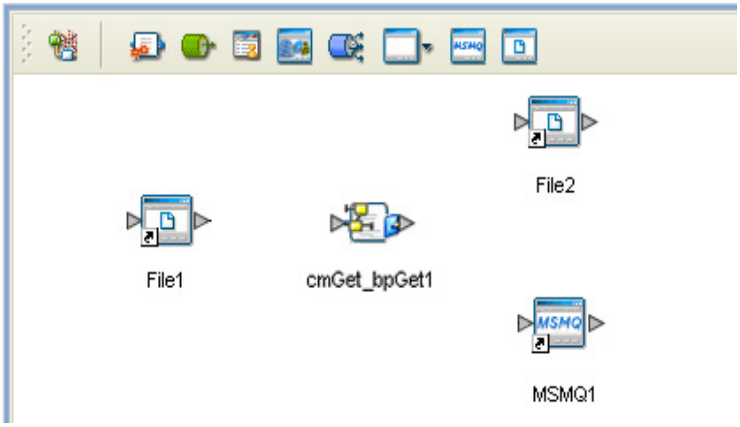
Add the Project components to the Connectivity Map by dragging the components you require from the toolbar or Project Explorer tree, to the canvas.

1 For this sample, drag and drop the following components onto the canvas as displayed in **Figure 21 on page 54**:

- 1 **MSMQ** External Application (from the Connectivity Map toolbar)

- 2 **File** External Applications (from the Connectivity Map toolbar)
- **bpGet** Business Process (from the Business Rules tree)

Figure 21 cmGet Connectivity Map



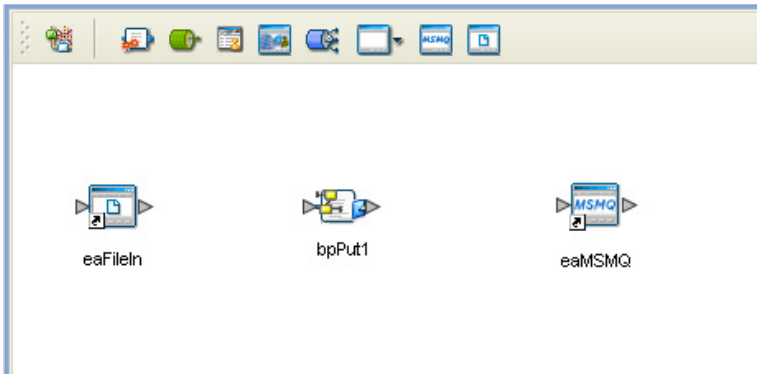
2 Rename the cmGet Connectivity Map's components as follows:

- **File1** to **eaFileIn**
- **File2** to **eaFileOut**
- **MSMQ1** to **eaMSMQ**
- **cmGet_bpGet1** to **bpGet1**

3 Create the **cmPut** Connectivity Map in the same manner as above. To populate the cmPut Connectivity Map, drag the following components from the Project Explorer tree to your Connectivity Map canvas as displayed in Figure 22:

- **eaFileIn** External Application
- **eaMSMQ** External Application
- **bpPut** Business Process (rename cmPut_bpPut1 to **bpPut1**)

Figure 22 cmPut Connectivity Map

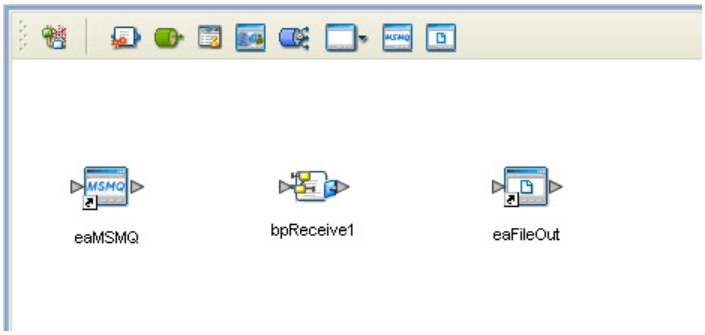


4 Create the **cmReceive** Connectivity Map in the same manner as above. To populate the **cmReceive** Connectivity Map, drag the follow components from the Project Explorer tree to your Connectivity Map canvas as displayed in Figure 23:

- **eaFileIn** External Application

- **eaMSMQ** External Application
- **bpReceive** Business Process (rename cmReceive_bpReceive1 to **bpReceive1**)

Figure 23 cmReceive Connectivity Map

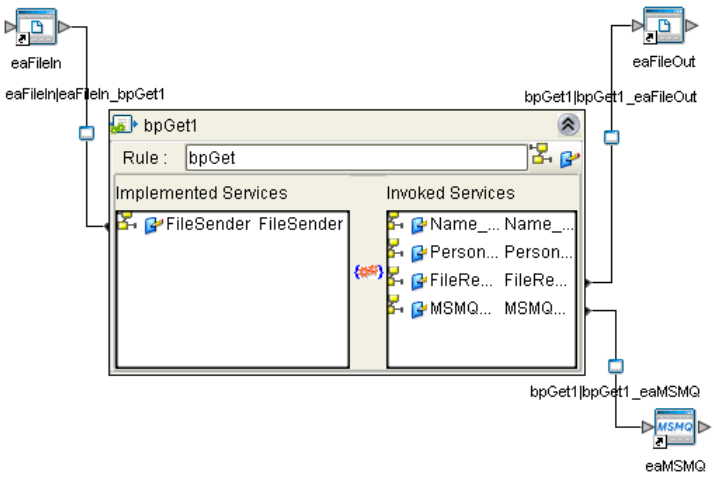


4.5.6 Binding the Project Components

Components are associated and the bindings are created from the Connectivity Maps.

1. From the **cmGet** Connectivity Map, double-click the **bpGet1** service. The **bpGet1** binding box appears with the **bpGet** Rule.
2. From the **bpGet1** binding box, map **FileSender** (under Implemented Services) to the **eaFileIn** application. To do this, click on FileSender (under Implemented Services) and drag your cursor to the output node of the **eaFileIn** application. A link now appears between the two components.
3. Map **MSMQReceiver** (under Invoked Services) to the **eaMSMQ** application.
4. Map **FileReceiver** (under Invoked Services) to the **eaFileOut** application (see Figure 24).

Figure 24 Connectivity Map - Binding the Components



5. Minimize the **bpGet1** binding box by clicking the chevrons in the upper-right corner.

Binding the components of the cmPut Connectivity Map

1. From the **cmPut Connectivity Map**, double-click the **bpPut1** service. The **bpPut1** Binding dialog box appears using **bpPut** as the rule.
2. From the **bpPut1** Binding dialog box, drag **FileSender** (Implemented Services) to the **eaFileIn** External Application.
3. From the **bpPut1** Binding dialog box, drag **MSMQReceiver** (under Invoked Services) to the **eaMSMQ** External Application.
4. Minimize the **bpPut1** binding box.

Binding the components of the cmPut Connectivity Map

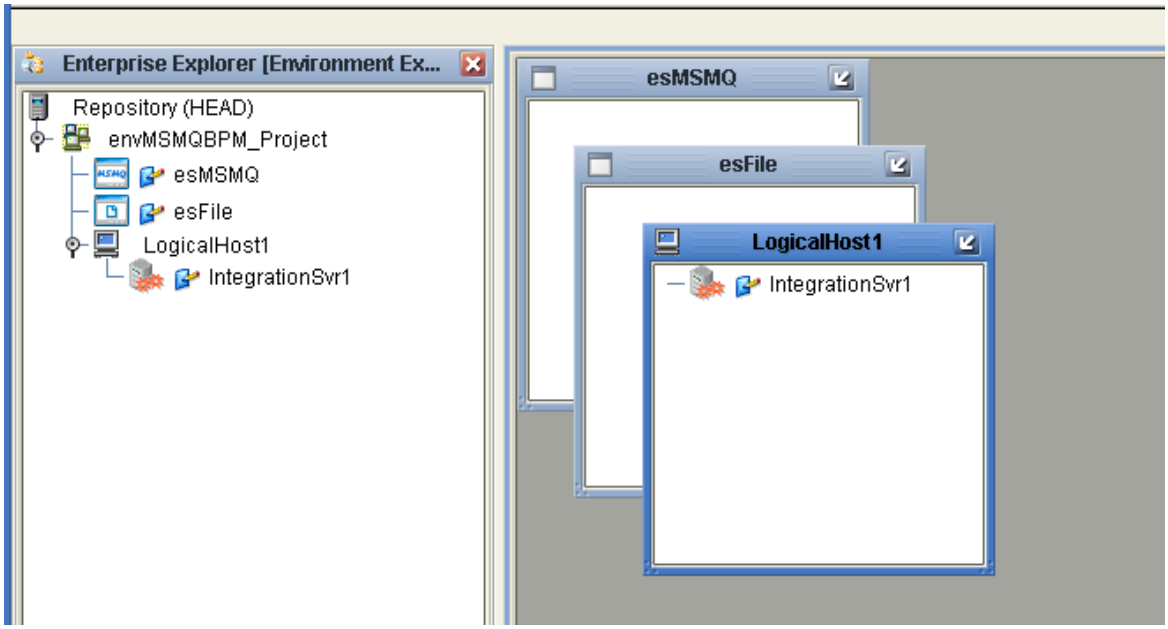
1. From the **cmReceive Connectivity Map**, double-click the **bpReceive1** service. The **bpReceive1** Binding dialog box appears using **bpReceive** as the rule.
2. From the **bpReceive1** Binding dialog box, drag **MSMQSender** (under Implemented Services) to the **eaMSMQ** External Application.
3. From the **bpReceive1** Binding dialog box, drag **FileReceiver** (under Invoked Services) to the **eaFileOut** External Application.
4. Save your current changes to the Repository.

4.5.7 Creating an Environment

Environments include the external systems, Logical Hosts, integration servers and message servers used by a Project and contain the configuration information for these components.

1. From the Enterprise Designer's Enterprise Explorer, click the **Environment Explorer** tab.
2. Right-click the Repository and select **New Environment**. A new Environment is added to the Environment Explorer tree.
3. Rename the new Environment to **envMSMQBPM**.
4. Right-click **envMSMQBPM** and select **New > MSMQ External System**. Name the External System **esMSMQ** and click **OK**. **esMSMQ** is added to the Environment Editor.
5. Right-click **envMSMQBPM** and select **New > File External System**. Name the External System **esFile** and click **OK**. **esFile** is added to the Environment Editor.
6. Right-click **envMSMQBPM** and select **New > Logical Host**. The **LogicalHost1** box is added to the Environment.
7. Right-click **LogicalHost1** and select **New > Sun SeeBeyond Integration Server**. A new Integration Server (**IntegrationSvr1**) is added to the Environment Explorer tree under LogicalHost1 (see Figure 25).

Figure 25 Environment Editor - Env_MSMQ Environment



8 Save your current changes to the Repository.

4.5.8 Configuring the eWays

Connectivity Map properties are specific for each eWay. Configure the eWay properties for your system, including the following settings:

cmGet Connectivity Map Properties

- **eaFileIn** properties:
  - Parameter Settings > Input file name: **Trigger.txt**
- **eaFileOut** properties:
  - Parameter Settings > Output file name: **outputMSMQBPget%d.dat**
- **eaMSMQ** properties:
  - MSMQ Configuration: Specify your queue using the MSMQ Queue Alias, MSMQ Format Name or MSMQ Queue Name. See **"Outbound MSMQ eWay Connectivity Map Properties" on page 27** for more details.
  - MSMQ Configuration > MSMQ Access Mode: **SEND_ACCESS**

cmPut Connectivity Map Properties

- **eaFileIn** properties:
  - Parameter Settings > Input file name: **Name.xml**
- **eaMSMQ** properties:

- MSMQ Configuration: Specify your queue using the MSMQ Queue Alias, MSMQ Format Name or MSMQ Queue Name. See **"Outbound MSMQ eWay Connectivity Map Properties" on page 27** for more details.
- MSMQ Configuration > MSMQ Access Mode: **SEND_ACCESS**

cmReceive Connectivity Map Properties

- **eaFileOut** properties:
  - Parameter Settings > Output file name: **outputMSMQBPRcv%d.dat**
- **eaMSMQ** properties:
  - MSMQ Configuration: Specify your queue using the MSMQ Queue Alias, MSMQ Format Name or MSMQ Queue Name. See **"Outbound MSMQ eWay Connectivity Map Properties" on page 27** for more details.
  - MSMQ Configuration > MSMQ Access Mode: **RECEIVE_ACCESS**

Environment Properties

Environment properties are global for all eWays that share an External System.

- **esFile**
  - Inbound File eWay > Parameter Settings > Directory: *your input directory*
  - Outbound File eWay > Parameter Settings > Directory: *your output directory*
- **esMSMQ**
  - Inbound MSMQ eWay > MSMQ Environment > MSMQ Host Name: *Your MSMQ Host Name*
  - Outbound MSMQ eWay > MSMQ Environment > MSMQ Host Name: *Your MSMQ Host Name*

See **"Configuring the MSMQ eWay" on page 18** for more information about configuring the MSMQ eWays for your system.

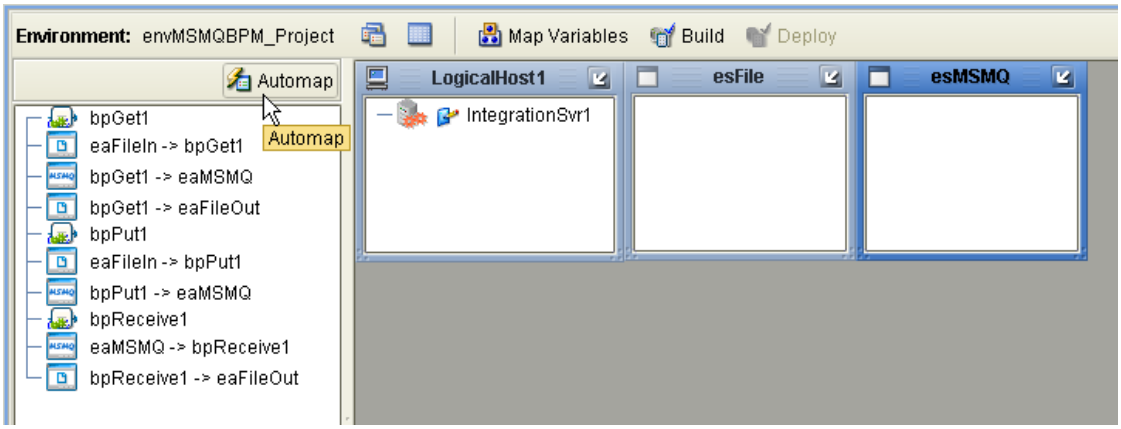
4.5.9 Creating and Activating the Deployment Profile

Deployment Profiles assign Collaborations and Message Destinations to the Integration Server and Message Server. You create a Deployment Profiles using the Deployment Editor. To create the sample Deployment Profile, do the following.

1. From the Enterprise Explorer's Project Explorer, right-click the Project (**prjMSMQBPM**) and select **New** > **Deployment Profile**. The **Create Deployment Profile** dialog box appears.
2. From the **Create Deployment Profile** dialog box, enter a name for the Deployment Profile (for this sample, **dpMSMQBPM**). Make sure that the selected Environment is **envMSMQBPM** and that all of your Connectivity Maps are selected. Click **OK**. The **Deployment Editor** appears.

3 From the Deployment Editor toolbar, click the **Auto Map** icon (see Figure 26). The Project's components are automatically mapped to their system windows.

Figure 26 Deployment Profile - Auto Map



4 Save your changes to the Repository.

4.5.10 Configuring the Integration Server

You must set your Sun SeeBeyond Integration Server Password property before deploying your Project.

1 From the Environment Explorer, right-click **IntegrationSvr1** under your **Logical Host**, and select **Properties** from the shortcut menu. The Integration Server Properties Editor appears.

2 Click the **Password** property field under **Sun SeeBeyond Integration Server Configuration**. An ellipsis appears in the property field.

3 Click the ellipsis. The **Password Settings** dialog box appears. Enter **STC** as the **Specific Value** and as the **Confirm Password**, and click **OK**.

4 Click **OK** to accept the new property and close the Properties Editor.

4.5.11 Creating and Starting the Domain

To deploy your Project, you must first create a domain. A domain is an instance of a Logical Host.

Create and Start the Domain

1 Navigate to your **<*JavaCAPS51*>\logicalhost** directory (where <*JavaCAPS51*> is the location of your Sun Java Composite Application Platform Suite installation.

2 Double-click the **domainmgr.bat** file. The **Domain Manager** appears.

3 If you have already created a domain, select your domain in the Domain Manager and click the **Start an Existing Domain** button. Once your domain is started, a green check mark indicates that the domain is running.

4 If there are no existing domains, a dialog box indicates that you can create a domain now. Click **Yes**. The **Create Domain** dialog box appears.

5 Make any necessary changes to the **Create Domain** dialog box and click **Create**. The new domain is added to the Domain Manager. Select the domain and click the **Start an Existing Domain** button. Once your domain is started, a green check mark indicates that the domain is running.

For more information about creating and managing domains see the *Sun SeeBeyond eGate Integrator System Administration Guide*.

4.5.12 Building and Deploying the Project

The Build process compiles and validates the Project's Java files and creates the Project EAR file.

Build the Project

1 From the Deployment Editor toolbar, click the **Build** icon.

2 If there are any validation errors, a **Validation Errors** pane will appear at the bottom of the Deployment Editor and displays information regarding the errors. Make any necessary corrections and click **Build** again.

3 After the Build has succeeded you are ready to deploy your Project.

Deploy the Project

1 From the Deployment Editor toolbar, click the **Deploy** icon. Click **Yes** when the **Deploy** prompt appears.

2 A message appears when the project is successfully deployed. You can now test your sample.

Note: *Projects can also be deployed from the Enterprise Manager. For more information about using the Enterprise Manager to deploy, monitor, and manage your projects, see the Sun SeeBeyond eGate™ Integrator System Administration Guide.*

4.5.13 Running the Sample

To run your deployed sample Project do the following

1 From your configured input directory, paste (or rename) the sample input file to trigger the eWay.

2 From your output directory, verify the output data.

Chapter 5

Implementing an MSMQ eWay Project using Java Collaboration Definitions

This chapter provides an introduction to the MSMQ eWay components and information about how these components are created and implemented in an eGate Project. It is assumed that the reader understands the basics of creating a Project using the SeeBeyond Enterprise Designer. For more information on creating an eGate Project, see the *Sun SeeBeyond eGate Integrator Tutorial* and the *Sun SeeBeyond eGate Integrator User's Guide*.

What's in This Chapter

- **MSMQ eWay Components** on page 61
- **MSMQ eWay Sample Project** on page 62
- **Importing a Sample Project** on page 62

5.1 MSMQ eWay Components

This chapter presents a sample MSMQ eWay Project, created using the same procedures as the sample end-to-end Project provided in the *eGate Tutorial*. The eWay components that are unique to the MSMQ eWay include the following:

- MSMQ eWay Properties file: The Properties file for the MSMQ eWay contains the parameters necessary to connect with a specific external system. These parameters are set with the Properties Editor. For more information about the MSMQ eWay Properties file and the Properties Editor, see **"Configuring the MSMQ eWay" on page 18**.
- MSMQClient OTD: The **MSMQClient** OTD is provided with the eWay and contains methods and attributes used to create the Business Rules that invoke the **MSMQ** program.

5.2 MSMQ eWay Sample Project

The MSMQ eWay includes a sample Project that uses eGate with Java Collaboration Definitions, named **prjMSMQJCD**. This sample Project performs three separate operations that demonstrate the following:

- Get Operation: The eWay subscribes to an MSMQ queue. The File eWay is used as a trigger. When a "trigger" message is received by the File eWay, the MSMQ eWay picks up any available XML data (messages) from the queue and publishes them to a Java Collaboration. The Collaboration unmarshals the message, modifies the data by converting a name from one value (first, last) to two values (first and last), and marshals the data. The message is then published to a outbound File eWay. The File eWay publishes the message an external directory.
- Put Operation: A File eWay subscribes to an external directory. XML data (message) is picked up from the input directory and published to a Java Collaboration. The Collaboration unmarshals the message, modifies the data converting a name from two values (first and last) to one value (first, last), and marshals the data. The message is then published to an MSMQ queue.
- **Receive Operation:** The eWay subscribes to an MSMQ queue. XML data (message) is picked up from the queue and published to a Java Collaboration. The Collaboration unmarshals the message, modifies the data converting a name from one value (first, last) to two values (first and last), and marshals the data. The message is then published to a outbound File eWay. The File eWay publishes the message to a directory.

Both the **Get** and **Receive** operations read the message from the queue. The **Get** operation requires a trigger, such as the presence of an input file. **Receive**, on the other hand, routinely polls the queue for messages. The polling rate is specified by the inbound property, **MSMQ Receive Interval**. No trigger is necessary.

This chapter provides directions for importing and completing the prjMSMQJCD. It also explains how to create the sample Project manually.

5.3 Importing a Sample Project

Sample eWay Projects are included as part of the installation package. To import a sample eWay Project to the Enterprise Designer do the following:

1. The sample files are uploaded with the eWay's documentation SAR file and downloaded from the Sun Java Composite Application Platform Suite Installer's **Documentation** tab. The **MSMQ_eWay_Sample.zip** file contains the various sample Project ZIP files. Extract the samples to a local file.
2. Save all unsaved work before importing a Project.
3. From the Enterprise Designer's Project Explorer pane, right-click your Repository and select **Import** from the shortcut menu. The **Import Manager** appears.

4 Browse to the directory that contains the sample Project zip file. Select the sample file (for this sample, **MSMQJCD_Sample.zip**) and click **Import**. After the sample Project is successfully imported, click **Close**.

5 Before an imported sample Project can be run you must do the following:

- Create an **Environment** (see **"Creating an Environment" on page 84**)
- Configure the eWays for your specific system (see **"Configuring the eWays" on page 84**)
- Create a **Deployment Profile** (see **"Creating the Deployment Profile" on page 86**)
- Create and start a domain (see **"Creating and Starting the Domain" on page 87**)
- Build and deploy the Project (see **"Building and Deploying the Project" on page 88**)

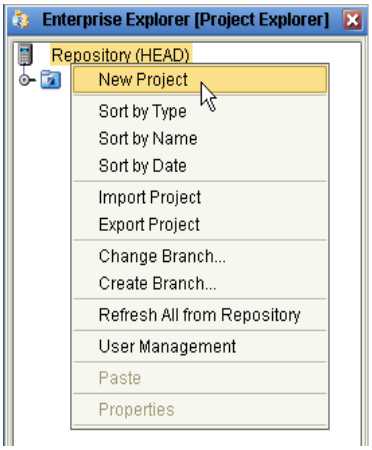
5.4 Creating the prjMSMQJCD Project

The following section explains how the sample Project's components are created. The first step is to create and name a new Project in eGate Enterprise Designer.

5.4.1 Creating a New Project

1 From the Enterprise Designer's Project Explorer tree, right-click your Repository and select **New Project.** A new Project (**Project1**) appears on the Project Explorer tree (see **Figure 27 on page 63**).

Figure 27 Enterprise Explorer - New Project



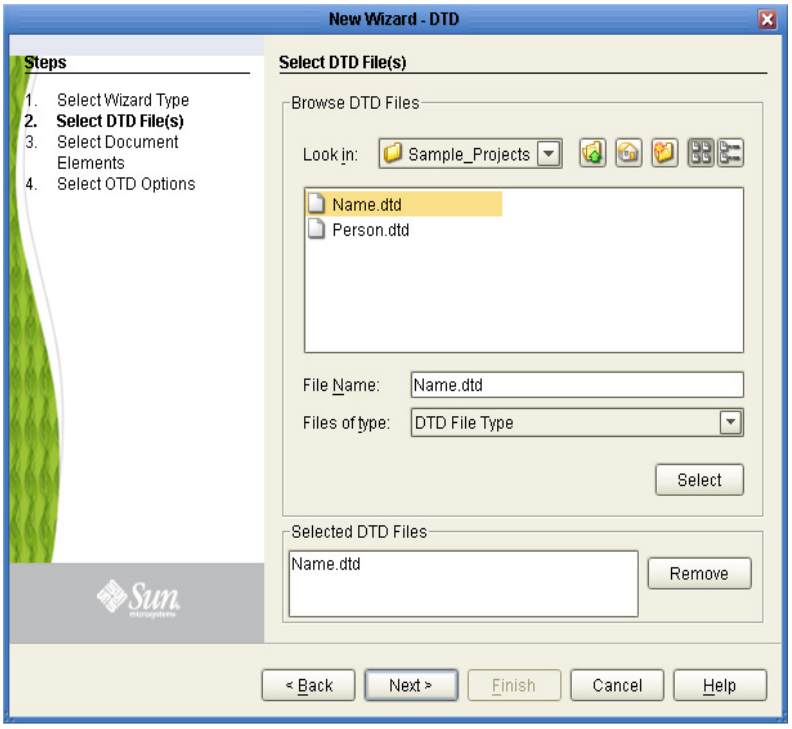
2 Rename **Project1** to **prjMSMQJCD**.

5.4.2 Creating OTDs Using the DTD Wizard

Two sample XML Document Type Definitions (DTDs) are included with the sample Projects: **Name.dtd** and **Person.dtd**. Convert these to **Object Type Definitions** (OTDs) using the **Object Type Definition Wizard**.

1. From the Project Explorer tree, right-click the **prjMSMQJCD** Project and select **New** > **Object Type Definition** from the shortcut menu. The **Object Type Definition Wizard** appears.
2. From the Select Wizard Type box, select **DTD**. The **DTD Wizard** appears.
3. For Step 2 of the wizard, "Select DTD File(s)," browse to the location where you saved the sample Project data, and select the Name.dtd file (see Figure 28) and click **Select**. Click **Next**.

Figure 28 DTD Wizard - Select DTD File(s)



4. For step 3 of the wizard, "Select Document Elements," click **Next**.
5. For step 4 of the wizard, "Select OTD Options," use the default settings and click **Finish**. The OTD Editor appears displaying the new **Name_Name.otd.**
6. Repeat steps 1 - 4 using **Person.dtd** to create the Person_Person.otd.
7. Save your work to the Repository and close the OTD Editor.

5.4.3 Creating the Collaboration Definitions

The next step to creating the sample is to create three Collaborations using the **Collaboration Definition Wizard (Java)**. These are:

- **jcdGet**: for the Get operation
- **jcdPut**: for the Put operation
- **jcdReceive**: for the Receive operation

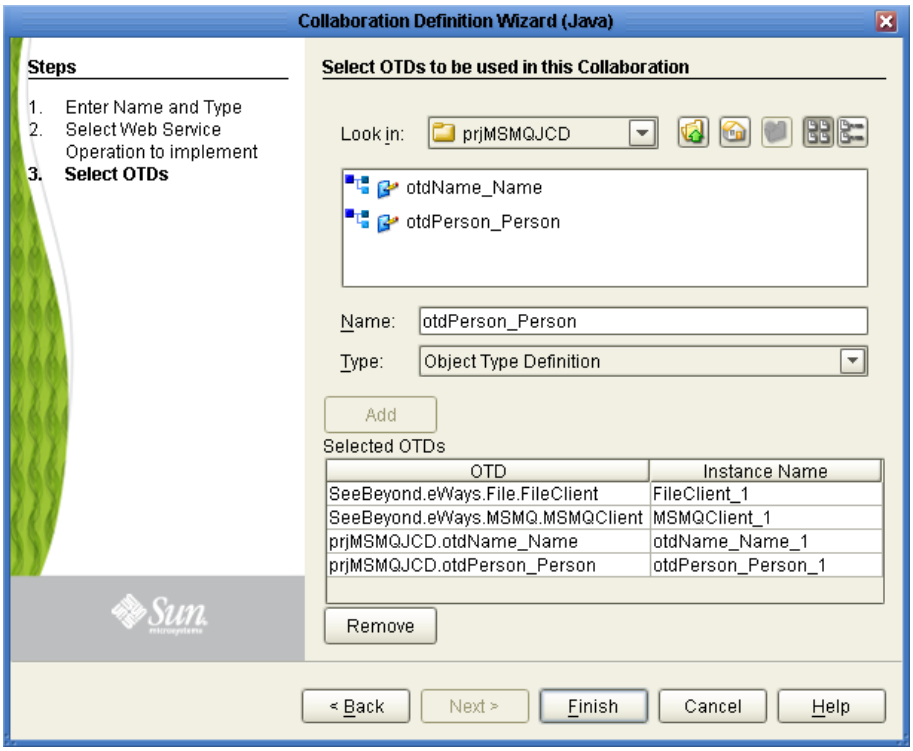
Once you have created these Collaboration Definitions, the Business Rules of the Collaborations can be written using the Collaboration Editor.

Create the jcdGet Collaboration

The **jcdGet** Collaboration defines transactions from an MSMQ queue using a File eWay as a trigger, to an outbound File eWay, and in process, modifies the data to convert a name from one value (first, last) to two values (first and last). To create the **jcdGet** Collaboration, do the following:

1. From the Project Explorer tree, right-click the **prjMSMQJCD** Project and select **New > Collaboration Definition (Java)** from the shortcut menu. The **Collaboration Definition Wizard** appears.
2. Enter a Collaboration name (for this sample **jcdGet)** and click **Next**.
3. In Step 2 of the wizard, "Select Web Service Operation to Implement," from the Web Services Interfaces selection window, select **Sun SeeBeyond > eWays > MSMQ > MSMQClient > getWSMSMQMessage**, and click **Next**.
4. In Step 3, "Select OTDs," click **Sun SeeBeyond > eWays > File** and double-click **FileClient**. The **FileClient_1** OTD is added to the Selected OTDs field.
5. Click the **Up One Level** button to return to the Repository level, and click **Sun SeeBeyond > eWays > MSMQ** and double-click **MSMQClient**. The **MSMQClient_1** OTD is added to the Selected OTDs field.
6. Click the **Up One Level** button to return to the Repository level, and click **prjMSMQJCD** and double-click both **Name_Name** and **Person_Person**. The **Name_Name_1** OTD and **Person_Person_1** OTD are added to the Selected OTDs field (see **Figure 29 on page 66**).

Figure 29 Collaboration Definition Wizard (Java) - Select OTDs Interface



7 Click **Finish**. The Collaboration Editor (Java) appears with the new **jcd_MQGet** Collaboration.

Create the jcdPut Collaboration

The **jcdPut** Collaboration defines transactions from a File eWay to an MSMQ queue, and in process, modifies the data to convert a name from one value (first, last) to two values (first and last). To create the **jcdPut** Collaboration, do the following:

Follow the steps for creating the **jcdGet** Collaboration, with the following changes:

A Enter **jcdPut** as the name of the new Collaboration.

B Select **Sun SeeBeyond > eWays > MSMQClient > putWSMSMQMessage** as the Web Service Operation (wizard Step 2).

C For Step 3 of the wizard, click the **Up One Level** button to return to the Repository level, and click **Sun SeeBeyond > eWays > MSMQ** and double-click **MSMQClient**. The **MSMQClient_1** OTD is added to the Selected OTDs field.

D Click the **Up One Level** button to return to the Repository level, and click **prjMSMQJCD** and double-click both **Name_Name** and **Person_Person**. The **Name_Name_1** OTD and **Person_Person_1** OTD are added to the Selected OTDs field.

E Click **Finish**. The Collaboration Editor (Java) appears with the new **jcdPut** Collaboration in the right pane of the Enterprise Designer.

Create the jcdReceive Collaboration

The **jcdReceive** Collaboration defines transactions from an MSMQ queue to a File eWay, and in process, modifies the data to convert a name from two values (first and last) to one value (first, last). To create the **jcdReceive** Collaboration, do the following:

Follow the steps for creating the **jcdGet** Collaboration, with the following changes:

A Enter **jcdReceive** as the name of the new Collaboration (wizard step 1).

B Select **Sun SeeBeyond > eWays > MSMQClient > receiveMSMQMessage** as the Web Service Operation (wizard step 2).

C For Step 3 of the wizard, click **Sun SeeBeyond > eWays > File** and double-click **FileClient**. The **FileClient_1** OTD is added to the Selected OTDs field.

D Click the **Up One Level** button to return to the Repository level, and click **prjMSMQJCD** and double-click both **Name_Name** and **Person_Person**. The **Name_Name_1** OTD and **Person_Person_1** OTD are added to the Selected OTDs field

E Click **Finish**. The Collaboration Editor (Java) appears with the new **jcdReceive** Collaboration in the right pane of the Enterprise Designer.

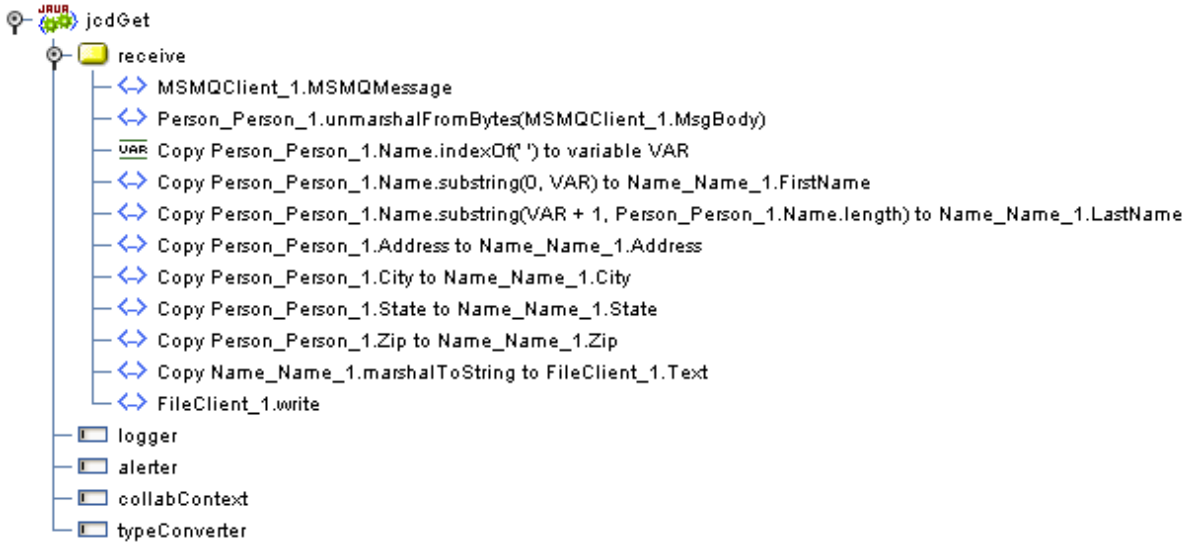
5.4.4 Creating the Business Rules Using the Collaboration Editor

The Business Rules for the **jcdGet** Collaboration are created using the Collaboration Editor (Java). To open the jcdGet Collaboration in the Collaboration Editor, from the Project Explorer tree, double-click the **jcd_MQGet** Collaboration. The Collaboration Editor opens to the selected Collaboration.

Creating the jcdGet Collaboration Business Rules

The **jcdGet** Collaboration contains the Business Rules displayed in Figure 30.

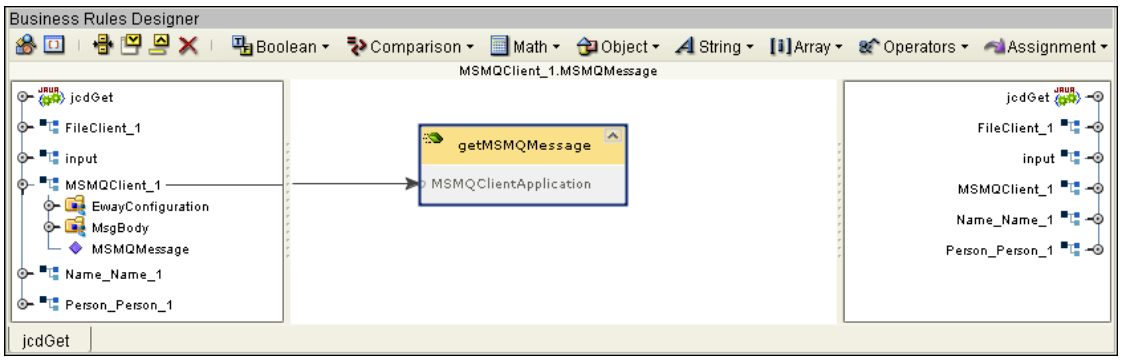
Figure 30 jcdGet Collaboration Business Rules



To create the jcdGet Collaboration Business Rules, do the following:

1. To create comments for the Business Rules, click the comment icon on the Business Rules toolbar. The **Enter a Comment** dialog box appears. Enter the comment and click **OK**. The comment is placed on the Business Rules tree under the last selected item. Once the comment is created, it can be moved by clicking the comment and dragging it up or down the Business Rules tree to a new location.
2. Create the **MSMQClient_1.MSMQMessage** Business Rule:
   - A From the Business Rules toolbar, click the **Rule** icon to add a new rule to the Business Rules tree.
   - B Right-click the **MSMQClient_1** node in the left pane of the Business Rule Designer, and click **Select method to call**. The method selection menu appears.
   - C Select **getMSMQMessage()** from the method selection menu. The **getMSMQMessage** method box appears with a link connecting **MSMQMessage_1**, in the left pane of the Business Rule Designer, to the **MSMQClientApplication** input node (see Figure 31).

Figure 31 Collaboration Editor - Business Rules Designer



D This creates the following Java code as displayed in the Java Source Editor:
`MSMQClient_1.getMSMQMessage();`

3 Create the **Person_Person_1.unmarshalFromBytes(MSMQClient_1.MsgBody)** Business Rule:

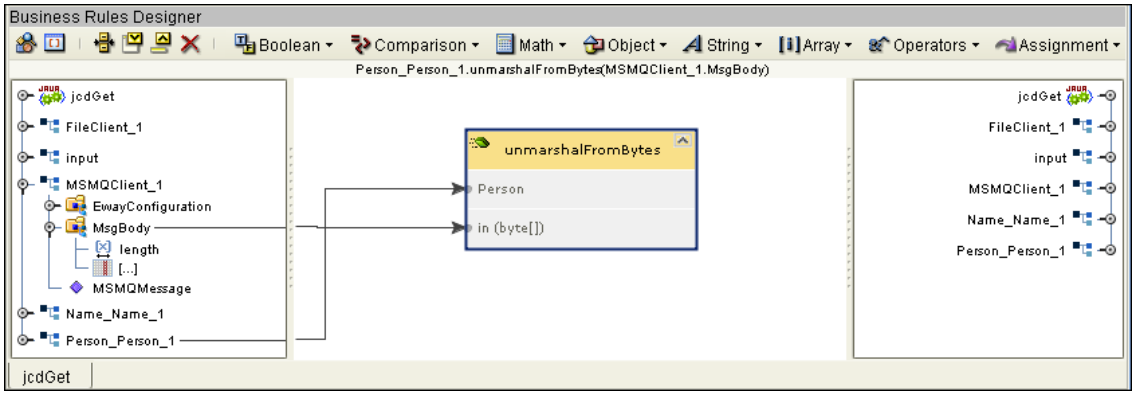
A From the Business Rules toolbar click the **rule** icon to add a new rule.

B Right-click the **Person_Person_1** node in the left pane of the Business Rule Designer, and click **Select method to call**. The method selection menu appears.

C Select **unmarshalFromBytes*(byte[] in)*** from the method selection menu. The **unmarshalFromBytes** method box appears with an arrow connecting **Person_Person_1**, in the left pane of the Business Rule Designer, to the **Person** input node.

D From the left pane of the Business Rules Designer, map **MsgBody**, under MSMQClient_1, to the in(byte[]) input node of the **unmarshalFromBytes** method box. To do this, click on **MsgBody** in the left pane of the Business Rules Designer and drag the cursor to the **in (byte[])** input node of the unmarshalFromBytes method box (see **Figure 32 on page 69**).

Figure 32 Collaboration Editor - Business Rules Designer



4 Create the **Copy Person_Person_1.Name.indexOf(' ') to variable VAR** Business Rule:

A From the Business Rules toolbar, click **Local Variable**. The **Create Variable** dialog box appears.

B Enter **VAR** as the variable name, select **Primitive > int**, as the Type, and click **OK**.

C Right-click **Name**, under **Person_Person_1** in the left pane of the Business Rule Designer, and select **Select method to call** from the shortcut menu. The method selection menu appears.

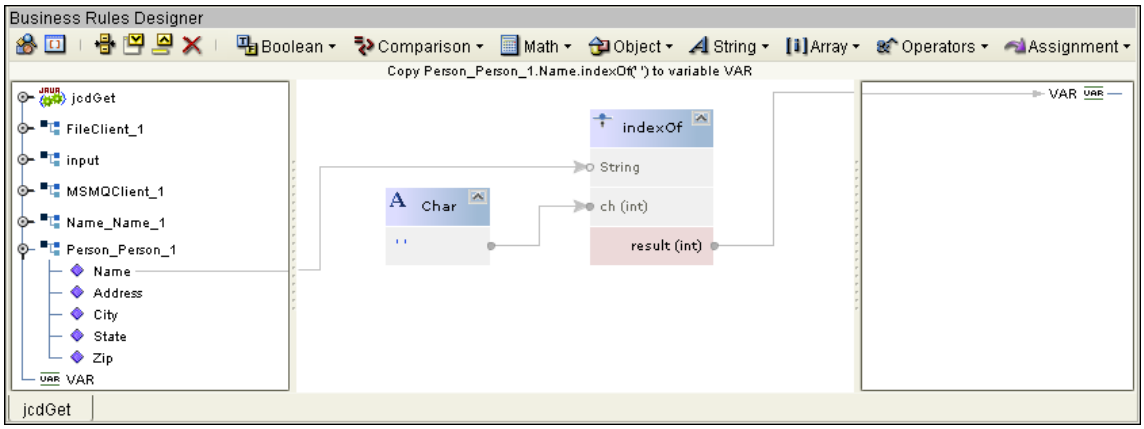
D Select **indexOf*(int ch)*** from the method selection menu. The **indexOf** method box appears.

E From the Business Rules Designer's **String** menu, select **Literal Character**. The **Char** literal box appears. Enter one space as the literal character value (press the space bar one time).

F Map the ***space*** output node of the **Char** literal box to the **ch (int)** input node of the **indexOf** method box.

G Map the **result (int)** output node of the **indexOf** method box to **VAR** in the right pane of the Business Rules Designer (see Figure 33).

Figure 33 Collaboration Editor - Business Rules Designer



5 Create the **Copy Person_Person_1.Name.substring(0, VAR) to Name_Name_1.FirstName** Business Rule:

A From the Business Rules toolbar, click the **rule** button to add a new rule.

B Right-click **Name** under **Person_Person_1** in the left pane of the Business Rule Designer, and click **Select method to call**. The method selection menu appears.

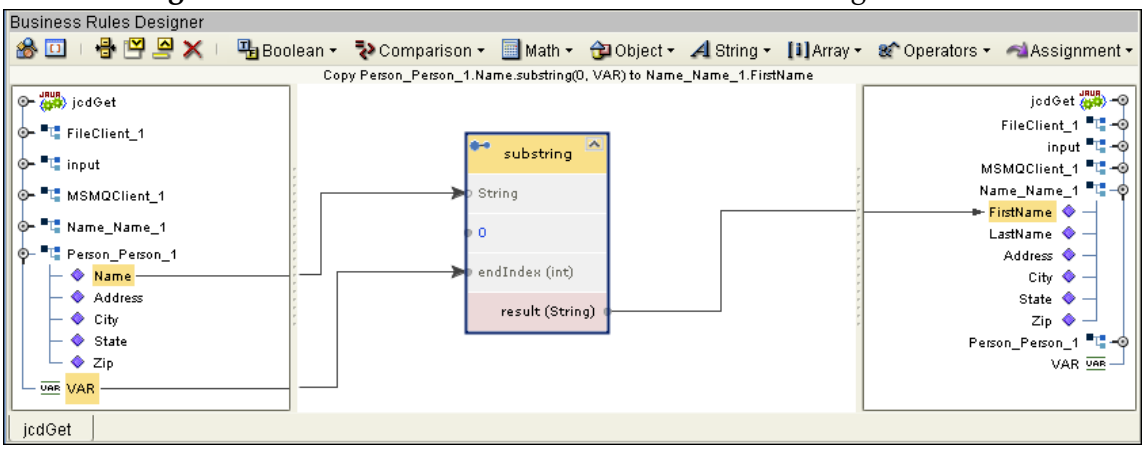
C Select **substring*(int beginIndex, int endIndex)*** from the method selection menu. The **substring** method box appears.

D Double-click the beginIndex field of the substring method box and enter **0** (zero) as the value.

E From the left pane of the Business Rules Designer, map **VAR** to the **endIndex (int)** input node of the **substring** method box.

F Map the **result (String)** output node of the substring method box to **FirstName**, under **Name_Name_1** in the right pane of the Business Rules Designer (see Figure 34).

Figure 34 Collaboration Editor - Business Rules Designer



G This creates the following Java code:

```
Name_Name_1.setFirstName(
Person_Person_1.getName().substring( 0, VAR ) );
```

6 Create the **Copy Person_Person_1.Name.substring(VAR + 1, Person_Person_1.Name.length) to Name_Name_1.LastName** Business Rule:

A From the Business Rules toolbar, click the **rule** button to add a new rule.

B Right-click **Name** under **Person_Person_1** in the left pane of the Business Rule Designer, and click **Select method to call**. The method selection menu appears.

C Select **substring*(int beginIndex, int endIndex)*** from the method selection menu. The **substring** method box appears with an arrow connecting **Name**, in the left pane of the Business Rule Designer, to the **String** input node.

D Again, right-click **Name** under **Person_Person_1** in the left pane of the Business Rule Designer, click **Select method to call**, and select **length()** as the method. The **length** method box appears.

E Right-click **VAR** in the left pane of the Business Rule Designer, click **Select method to call**, and select **Add** as the method. The **Add** method box appears.

F From the Business Rules Designer toolbar, select **Literal Number** from the Math menu. The **Number** Literal box appears with a value of **0**. Double-click the number and change the value to **1**.

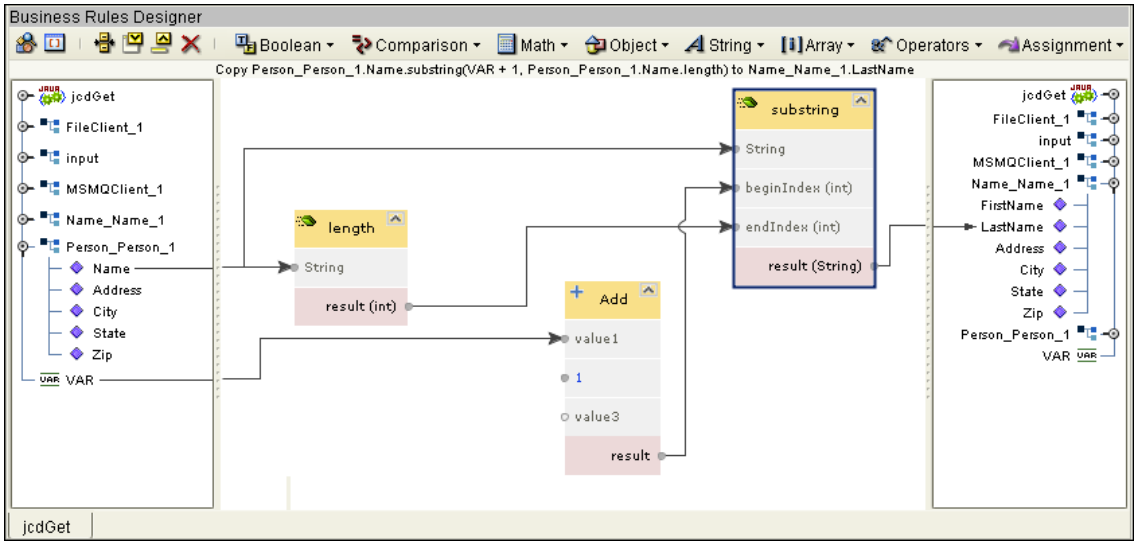
G Map the **1** output node of the Number literal box to the **value2** input node of the Add method box.

H Map the **result (int)** output node of the length box to the **endIndex (int)** input node of the substring method box.

I Map the **result** output node of the Add method box to the **beginIndex (int)** input node of the substring method box.

J Map the **result (String)** output node of the substring method box to **LastName** under **Name_Name_1** in the right pane of the Business Rules Designer (see Figure 35).

Figure 35 Collaboration Editor - Business Rules Designer



7 Create the **Copy Person_Person_1.Address to Name_Name_1.Address** Business Rule:

A From the Business Rules toolbar, click the **rule** button to add a new rule.

B Map **Address,** under **Person_Person_1** in the left pane of the Business Rules Designer, map, to **Address,** under **Name_Name_1**, in the right pane of the Business Rules Designer (see **Figure 36 on page 72**).

Figure 36 Collaboration Editor - Business Rules Designer

8 Create the **Copy Person_Person_1.City to Name_Name_1.City** Business Rule:

A From the Business Rules toolbar, click the **rule** button to add a new rule.

B Map **City,** under **Person_Person_1** in the left pane of the Business Rules Designer, map, to **City,** under **Name_Name_1**, in the right pane of the Business Rules Designer.

9 Create the **Copy Person_Person_1.State to Name_Name_1.State** Business Rule:

A From the Business Rules toolbar, click the **rule** button to add a new rule.

B Map **State,** under **Person_Person_1** in the left pane of the Business Rules Designer, map, to **State,** under **Name_Name_1**, in the right pane of the Business Rules Designer.

10 Create the **Copy Person_Person_1.Zip to Name_Name_1.Zip** Business Rule:

A From the Business Rules toolbar, click the **rule** button to add a new rule.

B Map **Zip,** under **Person_Person_1** in the left pane of the Business Rules Designer, map, to **Zip,** under **Name_Name_1**, in the right pane of the Business Rules Designer.

11 Create the **Copy Name_Name_1.marshalToString to FileClient_1.Text** Business Rule:

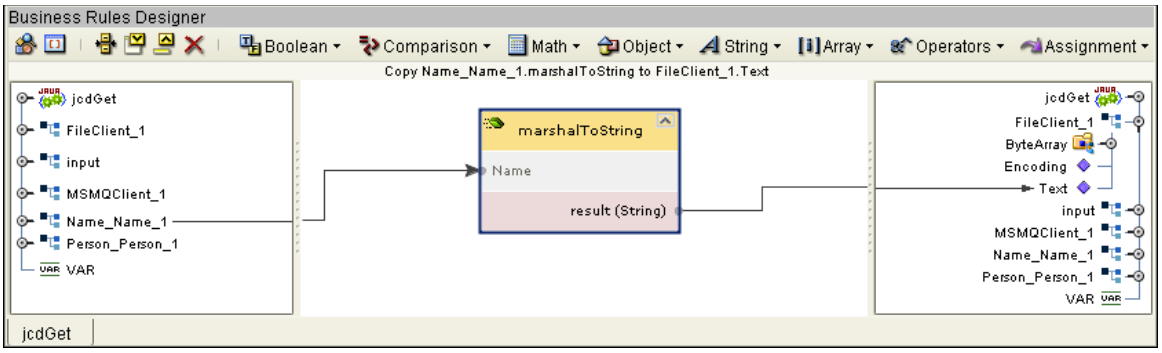
A From the Business Rules toolbar, click the **rule** button to add a new rule.

B Right-click **Name_Name_1** in the left pane of the Business Rule Designer, and click **Select method to call**. The method selection menu appears.

C Select **marshalToString()** from the method selection menu. The **marshalToString** method box appears with an arrow connecting **Name_Name_1**, in the left pane of the Business Rule Designer, to the **Name** input node.

D Map the **result (String)** output node of the marshalToString method box to **Text** under FileClient_1 in the right pane of the Business Rules Designer (see Figure 37).

Figure 37 Collaboration Editor - Business Rules Designer



12 Create the **FileClient_1.write** Business Rule:

A From the Business Rules toolbar, click the **rule** button to add a new rule.

B Right-click **FileClient_1** in the left pane of the Business Rule Designer, and click **Select method to call**. The method selection menu appears.

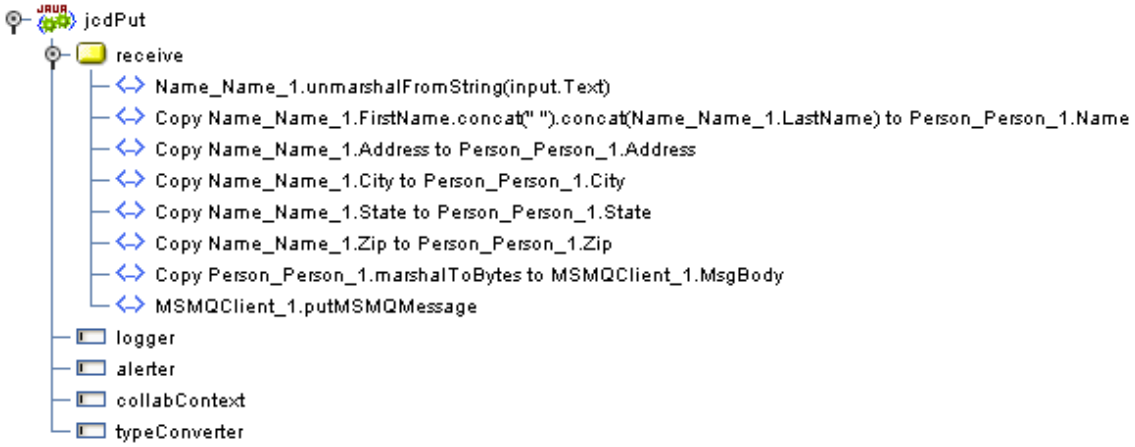
C Select **write()** from the method selection menu. The **write** method box appears.

13 Save your current changes to the Repository.

Creating the jcdPut Collaboration Business Rules

The jcdPut Collaboration contains the Business Rules displayed in Figure 38.

Figure 38 jcdPut Collaboration Business Rules



To create the jcdPut Collaboration Business Rules do the following:

1 Create the **Name_Name_1.unmarshalFromString(input.Text)** Business Rule:

A Right-click the **Name_Name_1** node in the left pane of the Business Rule Designer, and click **Select method to call**. The method selection menu appears.

B Select **unmarshalFromString*(String in)*** from the method selection menu. The **unmarshalFromString** method box appears.

C Map **Text**, under the **input** in the left pane of the Business Rules Designer, to the **in (String)** input node of the **unmarshalFromString** method box.

2 Create the **Copy Name_Name_1.FirstName.concat(" ").concat (Name_Name_1.LastName) to Person_Person_1.Name** Business Rule:

A From the Business Rules toolbar, click the **rule** button to add a new rule.

B From the Business Rules Designer's **String** menu, select **concat**. The **concat** method box appears. Repeat this step to create two **concat** method boxes on the Business Rules Designer canvas.

C Map **FirstName**, under **Name_Name_1 in** the left pane of the Business Rules Designer, to the **String** input node of the first **concat** method box.

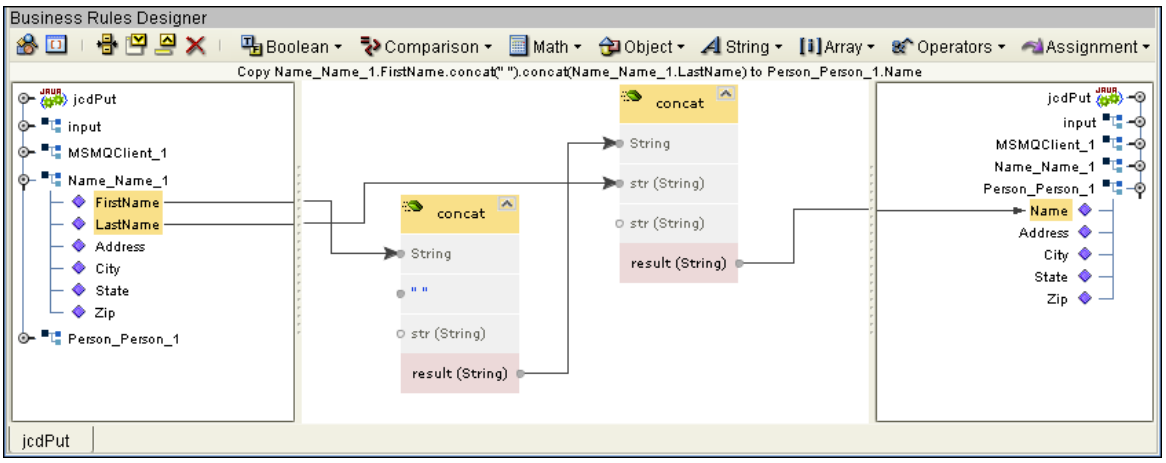
D Double-click str (String) field of the first concat method box, and enter one space (press space bar one time) as the literal String value.

E Map the **result (String)** output node of the first **concat** method box to the **String** input node of the second **concat** method box.

F Map **LastName** under **Name_Name_1** in the left pane of the Business Rules Designer, to the **str (String)** input node of the second **concat** method box.

G Map the **result (String)** output node of the second **concat** method box to **Name**, under the **Person_Person_1** node in the right pane of the Business Rules Designer (see Figure 39).

Figure 39 Collaboration Editor - Business Rules Designer



3 Create the **Copy Name_Name_1.Address to Person_Person_1.Address** Business Rule:

A From the Business Rules toolbar, click the **rule** button to add a new rule.

B Map **Address** under **Name_Name_1** in the left pane of the Business Rules Designer, to **Address**, under **Person_Person_1** in the right pane of the Business Rules Designer.

4 Create the **Copy Name_Name_1.City to Person_Person_1.City** Business Rule:

A From the Business Rules toolbar, click the **rule** button to add a new rule.

B Map **City** under **Name_Name_1** in the left pane of the Business Rules Designer, to **City**, under **Person_Person_1** in the right pane of the Business Rules Designer.

5 Create the **Copy Name_Name_1.State to Person_Person_1.State** Business Rule:

A From the Business Rules toolbar, click the **rule** button to add a new rule.

B Map **State** under **Name_Name_1** in the left pane of the Business Rules Designer, to **State**, under **Person_Person_1** in the right pane of the Business Rules Designer.

6 Create the **Copy Name_Name_1.Zip to Person_Person_1.Zip** Business Rule:

A From the Business Rules toolbar, click the **rule** button to add a new rule.

B Map **Zip** under **Name_Name_1** in the left pane of the Business Rules Designer, to **Zip**, under **Person_Person_1** in the right pane of the Business Rules Designer.

7 Create the **Copy Person_Person_1.marshalToBytes to MSMQClient_1.MsgBody** Business Rule:

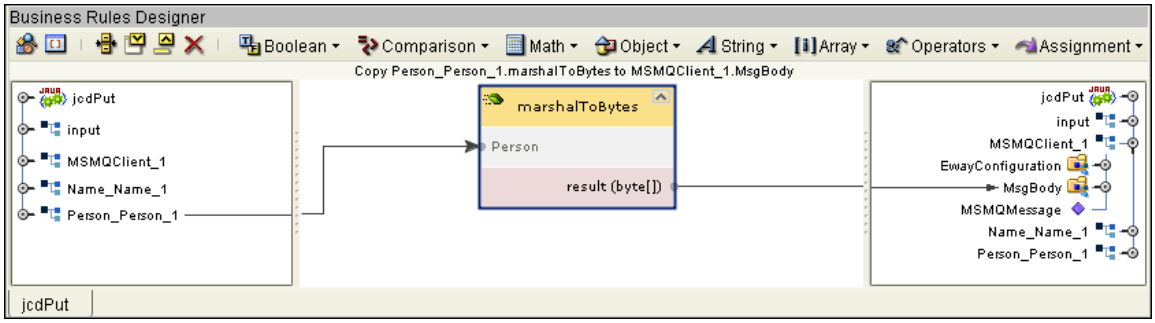
A From the Business Rules toolbar, click the **rule** button to add a new rule.

B Right-click **Person_Person_1** in the left pane of the Business Rule Designer, and click **Select method to call**. The method selection menu appears.

C Select **marshalToBytes()** from the method selection menu. The **marshalToBytes** method box appears.

D Map the **result (byte[])** output node of the **marshalToBytes** method box to **msgBody** under **MSMQClient_1** in the right pane of the Business Rules Designer.

Figure 40 Collaboration Editor - Business Rules Designer



8 Create the **MSMQClient_1.MSMQMessage** Business Rule:

A From the Business Rules toolbar, click the **rule** button to add a new rule.

B Right-click the **MSMQClient_1** node in the left pane of the Business Rule Designer, and click **Select method to call**. The method selection menu appears.

C Select **putMSMQMessage()** from the method selection menu. The **putMSMQMessage** method box appears.

9 Save your current changes to the Repository.

Creating the jcdReceive Collaboration Business Rules

The jcdReceive Collaboration contains the Business Rules displayed in Figure 41.

Figure 41 jcdReceive Collaboration Business Rules

To create the jcdReceive Collaboration Business Rules do the following:

1 Create the **If** statement and the **condition: input has Payload**:

   A From the Business Rules tree, right-click the **receiveMSMQMessage** node, and select **if/then** from the shortcut menu. An **If** statement is added to the Business Rules tree.

   B Right-click **input** in the left pane of the Business Rule Designer, and select **Select method to call** from the shortcut menu. The method selection box appears.

   C Select **hasPayload()** from the method selection menu. The **hasPayload** method box appears.

   D Map the **result (boolean)** output node of the **hasPayload** method box to **condition** in the right pane of the Business Rule Designer.

2 Create the **Person_Person_1.unmarshalFromBytes(input.Payload)** Business Rule under the **then** statement:

   A Select **then** under the **If** statement on the Business Rules tree, and from the Business Rules toolbar, click the **rule** button to add a new rule.

   B Right-click **Person_Person_1** in the left pane of the Business Rules Designer, and click **Select method to call**. The method selection menu appears.

   C Select **unmarshalFromBytes*(byte[] in)*** from the method selection menu. The **unmarshalFromBytes** method box appears.

   D Map **Payload** under the **input** node in the left pane of the Business Rules Designer, to the **in (byte[])** input node of the **unmarshalFromBytes** method box.

3 Create the **Copy Person_Person_1.Name.indexOf(' ') to variable VAR** Business Rule:

A From the Business Rules toolbar, click **Local Variable**. The **Create Variable** dialog box appears.

B Enter **VAR** as the variable name, select **Primitive > int**, as the Type, and click **OK**.

C Right-click **Name** under **Person_Person_1** in the left pane of the Business Rules Designer, and click **Select method to call**. The method selection menu appears.

D Select **indexOf*(int ch)*** from the method selection menu. The **indexOf** method box appears with an arrow connecting **Name**, in the left pane of the Business Rule Designer, to the **String** input node.

E From the Business Rules Designer's **String** menu, select **Literal Character**. The **Char** literal box appears. Enter one space as the literal character value (press the space bar one time).

F Map the ***space*** output node of the **Char** literal box to the **ch (int)** input node of the **indexOf** method box.

G Map the **result (int)** output node of the **indexOf** method box to **VAR** in the right pane of the Business Rules Designer (see **Figure 34 on page 71**).

4 Create the **Copy Person_Person_1.Name.substring(0, VAR) to Name_Name_1.FirstName** Business Rule:

A From the Business Rules toolbar, click the **rule** button to add a new rule.

B Right-click **Name** under **Person_Person_1** in the left pane of the Business Rule Designer, and click **Select method to call**. The method selection menu appears.

C Select **substring*(int beginIndex, int endIndex)*** from the method selection menu. The **substring** method box appears.

D Double-click the beginIndex field of the substring method box and enter **0** (zero) as the value.

E From the left pane of the Business Rules Designer, map **VAR** to the **endIndex (int)** input node of the **substring** method box.

F Map the **result (String)** output node of the substring method box to **FirstName**, under **Name_Name_1** in the right pane of the Business Rules Designer (see **Figure 34 on page 71**).

5 Create the **Copy Person_Person_1.Name.substring(VAR + 1, Person_Person_1.Name.length) to Name_Name_1.LastName** Business Rule:

A From the Business Rules toolbar, click the **rule** button to add a new rule.

B Right-click **Name** under **Person_Person_1** in the left pane of the Business Rule Designer, and click **Select method to call**. The method selection menu appears.

C Select **substring*(int beginIndex, int endIndex)*** from the method selection menu. The **substring** method box appears with an arrow connecting **Name**, in the left pane of the Business Rule Designer, to the **String** input node.

D Again, right-click **Name** under **Person_Person_1** in the left pane of the Business Rule Designer, click **Select method to call**, and select **length()** as the method. The **length** method box appears.

E Right-click **VAR** in the left pane of the Business Rule Designer, click **Select method to call**, and select **Add** as the method. The **Add** method box appears.

F From the Business Rules Designer toolbar, select **Literal Number** from the Math menu. The **Number** Literal box appears with a value of **0**. Double-click the number and change the value to **1**.

G Map the **1** output node of the Number literal box to the **value2** input node of the Add method box.

H Map the **result (int)** output node of the length box to the **endIndex (int)** input node of the substring method box.

I Map the **result** output node of the Add method box to the **beginIndex (int)** input node of the substring method box.

J Map the **result (String)** output node of the substring method box to **LastName** under **Name_Name_1** in the right pane of the Business Rules Designer (see **Figure 35 on page 72**).

6 Create the **Copy Person_Person_1.Address to Name_Name_1.Address** Business Rule:

A From the Business Rules toolbar, click the **rule** button to add a new rule.

B Map **Address,** under **Person_Person_1** in the left pane of the Business Rules Designer, map, to **Address,** under **Name_Name_1**, in the right pane of the Business Rules Designer **Figure 36 on page 72**).

7 Create the **Copy Person_Person_1.City to Name_Name_1.City** Business Rule:

A From the Business Rules toolbar, click the **rule** button to add a new rule.

B Map **City,** under **Person_Person_1** in the left pane of the Business Rules Designer, map, to **City,** under **Name_Name_1**, in the right pane of the Business Rules Designer.

8 Create the **Copy Person_Person_1.State to Name_Name_1.State** Business Rule:

A From the Business Rules toolbar, click the **rule** button to add a new rule.

B Map **State,** under **Person_Person_1** in the left pane of the Business Rules Designer, map, to **State,** under **Name_Name_1**, in the right pane of the Business Rules Designer.

9 Create the **Copy Person_Person_1.Zip to Name_Name_1.Zip** Business Rule:

A From the Business Rules toolbar, click the **rule** button to add a new rule.

B Map **Zip,** under **Person_Person_1** in the left pane of the Business Rules Designer, map, to **Zip,** under **Name_Name_1**, in the right pane of the Business Rules Designer.

10 Create the **Copy Name_Name_1.marshalToString to FileClient_1.Text** Business Rule:

A From the Business Rules toolbar, click the **rule** button to add a new rule.

B Right-click **Name_Name_1** in the left pane of the Business Rule Designer, and click **Select method to call**. The method selection menu appears.

C Select **marshalToString()** from the method selection menu. The **marshalToString** method box appears with an arrow connecting **Name_Name_1**, in the left pane of the Business Rule Designer, to the **Name** input node.

D Map the **result (String)** output node of the marshalToString method box to **Text** under FileClient_1 in the right pane of the Business Rules Designer (see **Figure 37 on page 73**).

11 Create the **FileClient_1.write** Business Rule:

A From the Business Rules toolbar, click the **rule** button to add a new rule.

B From the left pane of the Business Rules Designer, right-click **FileClient_1**, and click **Select method to call**. The method selection menu appears.

C Select **write()** from the method selection menu. The **write** method box appears.

12 Save your current changes to the Repository.

For information on how to create Business Rules using the Collaboration Editor, see the *Sun SeeBeyond eGate Integrator User's Guide*.

5.4.5 Creating the Connectivity Maps

Connectivity Maps provide a canvas for assembling and configuring a Project's components. The prjMSMQJCD Project uses three Connectivity Maps:

- **cmGet**: for the Get operation
- **cmPut**: for the Put operation
- **cmReceive**: for the Receive operation

To create the **cmGet** Connectivity Map, do the following:

1 From the Project Explorer, right-click **prjMSMQJCD** and select **New > Connectivity Map** from the shortcut menu.

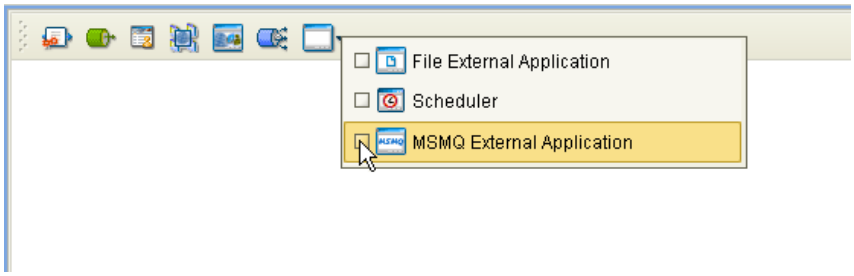
2 The New Connectivity Map appears and a node named **CMap1** is added to the Project on the Project Explorer tree. From the Project Explorer tree, rename the Connectivity Map to **cmGet**.

Select the External Applications

In a Connectivity Map, the icons on the toolbar represent available components used to populate the Connectivity Map canvas.

eWays in the Connectivity Map are associated with External Application. For example, to establish a connection to an MSMQ queue, you must first select MSMQ as an External Application to use in your Connectivity Map (see **Figure 42 on page 81**).

Figure 42 Connectivity Map - External Applications



1 Click the **External Application** icon on the Connectivity Map toolbar.

2 Select the External Applications that you require for your Project (for this sample, the **MSMQ**, and **File** External Applications). Icons representing the selected External Applications are added to the Connectivity Map toolbar.

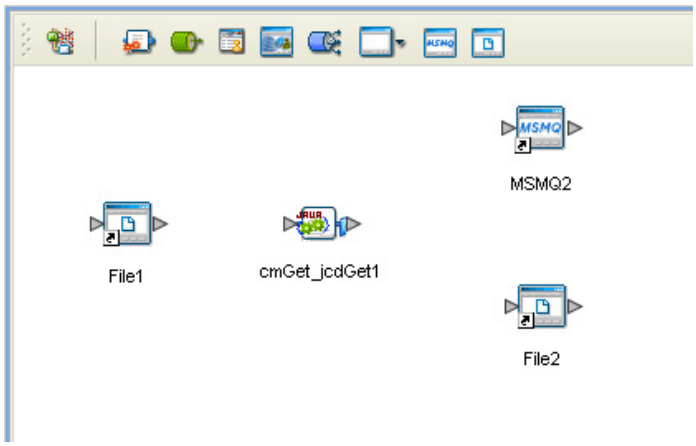
Populate the Connectivity Map

Add the Project components to the Connectivity Map by dragging the components you require from the toolbar or Project Explorer tree, to the canvas.

1 For this sample, drag and drop the following components onto the canvas as displayed in Figure 43:

- 1 **MSMQ** External Application (from the Connectivity Map toolbar)
- 2 **File** External Applications (from the Connectivity Map toolbar)
- **jcdGet** Business Process (from the Business Rules tree)

Figure 43 cmGet Connectivity Map



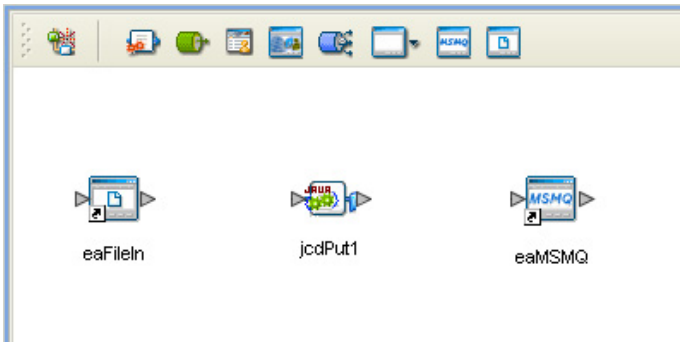
2 Rename the cmGet Connectivity Map's components as follows:

- **File1** to **eaFileIn**
- **File2** to **eaFileOut**
- **MSMQ1** to **eaMSMQ**
- **cmGet_jcdGet1** to **jcdGet1**

3 Create the **cmPut** Connectivity Map in the same manner as above. To populate the cmPut Connectivity Map, drag the following components from the Project Explorer tree to your Connectivity Map canvas as displayed in Figure 44:

- **eaFileIn** External Application
- **eaMSMQ** External Application
- **jcdPut** Business Process (rename **cmPut_jcdPut1** to **jcdPut1**)

Figure 44 cmPut Connectivity Map



4 Create the **cmReceive** Connectivity Map in the same manner as above. To populate the **cmReceive** Connectivity Map, drag the follow components from the Project Explorer tree to your Connectivity Map canvas as displayed in Figure 45:

- **eaFileIn** External Application
- **eaMSMQ** External Application
- **jcdReceive** Business Process (rename **cmReceive_jcdReceive1** to **jcdReceive1**)

Figure 45 cmReceive Connectivity Map

5.4.6 Binding the Project Components

Components are associated and the bindings are created from the Connectivity Maps.

1 From the **cmGet** Connectivity Map, double-click the **jcdGet1** service. The **jcdGet1** binding box appears with the **jcdGet** Rule.

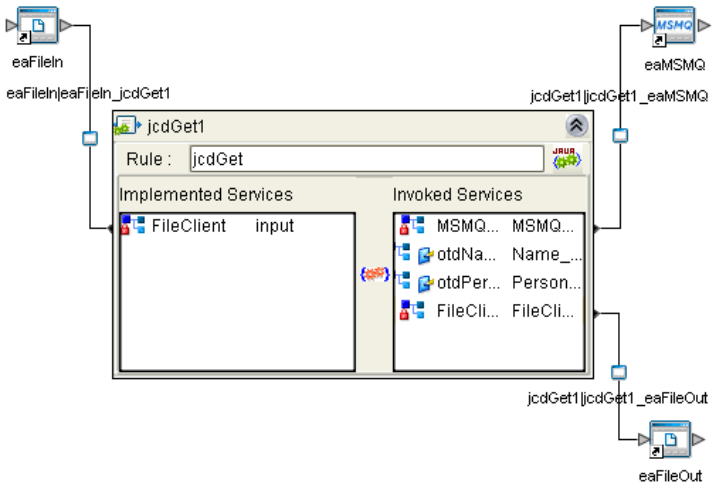
2 From the **jcdGet1** binding box, map **FileClient input** (under Implemented Services) to the **eaFileIn** application. To do this, click on **FileClient input** (under

Implemented Services) and drag your cursor to the output node of the **eaFileIn** application. A link now appears between the two components.

3 Map **MSMQClient_1** (under Invoked Services) to the **eaMSMQ** application.

4 Map **FileClient_1** (under Invoked Services) to the **eaFileOut** application (see Figure 46).

Figure 46 Connectivity Map - Binding the Components



5 Minimize the **jcdGet1** binding box by clicking the chevrons in the upper-right corner.

Binding the components of the cmPut Connectivity Map

1 From the **cmPut Connectivity Map**, double-click the **jcdPut1** service. The **jcdPut1** Binding dialog box appears using **jcdPut** as the rule.

2 From the **jcdPut1** Binding dialog box, drag **FileClient input** (Implemented Services) to the **eaFileIn** External Application.

3 From the **jcdPut1** Binding dialog box, drag **MSMQRClient_1** (under Invoked Services) to the **eaMSMQ** External Application.

4 Minimize the **jcdPut1** binding box.

Binding the components of the cmPut Connectivity Map

1 From the **cmReceive Connectivity Map**, double-click the **jcdReceive1** service. The **jcdReceive1** Binding dialog box appears using **jcdReceive** as the rule.

2 From the **jcdReceive1** Binding dialog box, drag **MSMQClient input** (under Implemented Services) to the **eaMSMQ** External Application.

3 From the **jcdReceive1** Binding dialog box, drag **FileClient_1** (under Invoked Services) to the **eaFileOut** External Application.

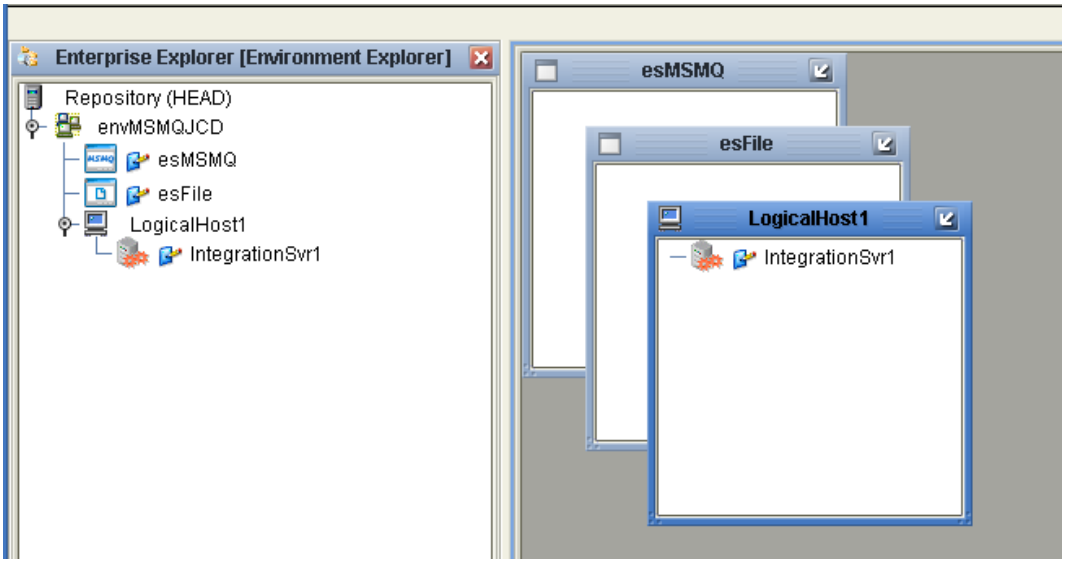
4 Save your current changes to the Repository.

5.4.7 Creating an Environment

Environments include the external systems, Logical Hosts, integration servers and message servers used by a Project and contain the configuration information for these components.

1. From the Enterprise Designer's Enterprise Explorer, click the **Environment Explorer** tab.
2. Right-click the Repository and select **New Environment**. A new Environment is added to the Environment Explorer tree.
3. Rename the new Environment to **envMSMQJCD**.
4. Right-click **envMSMQJCD** and select **New > MSMQ External System**. Name the External System **esMSMQ** and click **OK**. **esMSMQ** is added to the Environment Editor.
5. Right-click **envMSMQJCD** and select **New > File External System**. Name the External System **esFile** and click **OK**. **esFile** is added to the Environment Editor.
6. Right-click **envMSMQJCD** and select **New > Logical Host**. The **LogicalHost1** box is added to the Environment.
7. Right-click **LogicalHost1** and select **New > Sun SeeBeyond Integration Server**. A new Integration Server (**IntegrationSvr1**) is added to the Environment Explorer tree under LogicalHost1 (see Figure 47).

Figure 47 Environment Editor - Env_MSMQ Environment



8. Save your current changes to the Repository.

5.4.8 Configuring the eWays

Connectivity Map properties are specific for each eWay. Configure the eWay properties for your system, including the following settings:

cmGet Connectivity Map Properties

- **eaFileIn** properties:
  - Parameter Settings > Input file name: **Trigger.txt**
- **eaFileOut** properties:
  - Parameter Settings > Output file name: **outputMSMQBPget%d.dat**
- **eaMSMQ** properties:
  - MSMQ Configuration: Specify your queue using the MSMQ Queue Alias, MSMQ Format Name or MSMQ Queue Name. See **"Outbound MSMQ eWay Connectivity Map Properties" on page 27** for more details.
  - MSMQ Configuration > MSMQ Access Mode: **SEND_ACCESS**

cmPut Connectivity Map Properties

- **eaFileIn** properties:
  - Parameter Settings > Input file name: **Name.xml**
- **eaMSMQ** properties:
  - MSMQ Configuration: Specify your queue using the MSMQ Queue Alias, MSMQ Format Name or MSMQ Queue Name. See **"Outbound MSMQ eWay Connectivity Map Properties" on page 27** for more details.
  - MSMQ Configuration > MSMQ Access Mode: **SEND_ACCESS**

cmReceive Connectivity Map Properties

- **eaFileOut** properties:
  - Parameter Settings > Output file name: **outputMSMQBPRcv%d.dat**
- **eaMSMQ** properties:
  - MSMQ Configuration: Specify your queue using the MSMQ Queue Alias, MSMQ Format Name or MSMQ Queue Name. See **"Outbound MSMQ eWay Connectivity Map Properties" on page 27** for more details.
  - MSMQ Configuration > MSMQ Access Mode: **RECEIVE_ACCESS**

Environment Properties

Environment properties are global for all eWays that share an External System.

- **esFile**
  - Inbound File eWay > Parameter Settings > Directory: *your input directory*
  - Outbound File eWay > Parameter Settings > Directory: *your output directory*
- **esMSMQ**
  - Inbound MSMQ eWay > MSMQ Environment > MSMQ Host Name: *Your MSMQ Host Name*

- Outbound MSMQ eWay > MSMQ Environment > MSMQ Host Name: *Your MSMQ Host Name*

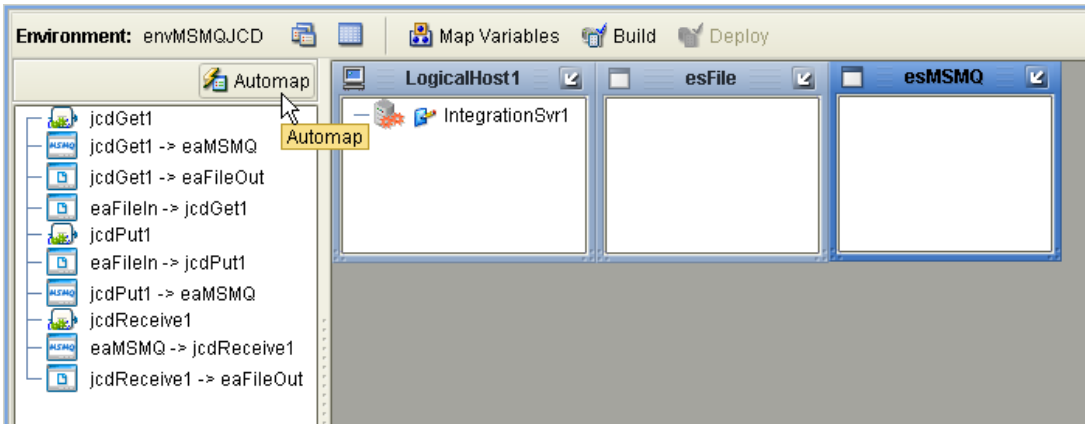
See **"Configuring the MSMQ eWay" on page 18** for more information about configuring the MSMQ eWays for your system.

5.4.9 Creating the Deployment Profile

Deployment Profiles assign Collaborations and Message Destinations to the Integration Server and Message Server. You create a Deployment Profiles using the Deployment Editor. To create the sample Deployment Profile, do the following.

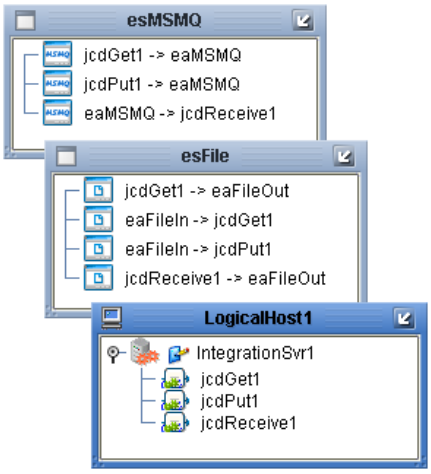
1. From the Enterprise Explorer's Project Explorer, right-click the Project (**prjMSMQJCD**) and select **New** > **Deployment Profile**. The **Create Deployment Profile** dialog box appears.
2. From the **Create Deployment Profile** dialog box, enter a name for the Deployment Profile (for this sample, **dpMSMQJCD**). Make sure that the selected Environment is **envMSMQJCD** and that all of your Connectivity Maps are selected. Click **OK**. The **Deployment Editor** appears.
3. From the Deployment Editor toolbar, click the **Automap** icon (see Figure 48). The Project's components are automatically mapped to their system windows.

Figure 48 Deployment Profile - Automap



The Project's components are automatically mapped to their system windows (see Figure 49).

Figure 49 Deployment Profile



4 Save your changes to the Repository.

5.4.10 Configuring the Integration Server

You must set your Sun SeeBeyond Integration Server Password property before deploying your Project.

1 From the Environment Explorer, right-click **IntegrationSvr1** under your **Logical Host**, and select **Properties** from the shortcut menu. The Integration Server Properties Editor appears.

2 Click the **Password** property field under **Sun SeeBeyond Integration Server Configuration**. An ellipsis appears in the property field.

3 Click the ellipsis. The **Password Settings** dialog box appears. Enter **STC** as the **Specific Value** and as the **Confirm Password**, and click **OK**.

4 Click **OK** to accept the new property and close the Properties Editor.

5.4.11 Creating and Starting the Domain

To deploy your Project, you must first create a domain. A domain is an instance of a Logical Host.

Create and Start the Domain

1 Navigate to your **<*JavaCAPS51*>\logicalhost** directory (where <*JavaCAPS51*> is the location of your Sun Java Composite Application Platform Suite installation.

2 Double-click the **domainmgr.bat** file. The **Domain Manager** appears.

3 If you have already created a domain, select your domain in the Domain Manager and click the **Start an Existing Domain** button. Once your domain is started, a green check mark indicates that the domain is running.

4 If there are no existing domains, a dialog box indicates that you can create a domain now. Click **Yes**. The **Create Domain** dialog box appears.

5 Make any necessary changes to the **Create Domain** dialog box and click **Create**. The new domain is added to the Domain Manager. Select the domain and click the **Start an Existing Domain** button. Once your domain is started, a green check mark indicates that the domain is running.

For more information about creating and managing domains see the *Sun SeeBeyond eGate Integrator System Administration Guide*.

5.4.12 Building and Deploying the Project

The Build process compiles and validates the Project's Java files and creates the Project EAR file.

Build the Project

1 From the Deployment Editor toolbar, click the **Build** icon.

2 If there are any validation errors, a **Validation Errors** pane will appear at the bottom of the Deployment Editor and displays information regarding the errors. Make any necessary corrections and click **Build** again.

3 After the Build has succeeded you are ready to deploy your Project.

Deploy the Project

1 From the Deployment Editor toolbar, click the **Deploy** icon. Click **Yes** when the **Deploy** prompt appears.

2 A message appears when the project is successfully deployed. You can now test your sample.

Note: *Projects can also be deployed from the Enterprise Manager. For more information about using the Enterprise Manager to deploy, monitor, and manage your projects, see the Sun SeeBeyond eGate™ Integrator System Administration Guide.*

5.4.13 Running the Sample

To run your deployed sample Project do the following

1 From your configured input directory, paste (or rename) the sample input file to trigger the eWay.

From your output directory, verify the output data.

Index

A

About **6**
ACTION_PEEK_CURRENT **30**
ACTION_RECEIVE **30**
Auto Map **59**, **86**

B

building a project **61**
Business Rule
  comments
    creating **68**
Business Rules
  Collaboration Editor (Java) **67**

C

Collaboration Definitions
  creating **64**
Collaboration Editor (Java) **67**
comments
  creating **68**
configuring the eWay properties **18**
configuring the MSMQ eWay **18**
Connection Mode **31**
Connectivity Map **53**, **80**
  populating **53**, **81**
conventions, text **8**

D

DENY_NONE **23**, **29**
DENY_RECEIVE_SHARE **23**, **29**
Deployment Profile
  Auto Map **59**, **86**
DLL files
  adding **15**
dynamic configuration **38**

E

eInsight
  engine and components **39**
  using with the WebSphere MQ eWay **39**
eWay
  configuring the eWays **57**, **84**
eWay components **61**
External Application
  selecting **18**, **19**
External Applications
  selecting **53**, **80**

I

implementation **61**

M

manual configuration **38**

O

operating systems
  requirements **10**
  supported **10**
organization of information **8**
overview
  MSMQ **6**

P

PEEK_ACCESS **30**
platforms
  requirements **10**
  supported **10**
Project
  importing **41**, **62**
project
  creating **41**, **63**
properties
  inbound Connectivity Map **22**
  inbound Environment **32**
  Max Pool Size **33**
  Message Priority
    outbound **29**
  MSMQ Access Mode
    inbound **24**
    outbound **30**
  MSMQ Format Name
    inbound **23**
  MSMQ Host Name
    inbound **32**, **35**
  MSMQ Queue Alias
    inbound **22**
    outbound **27**
  MSMQ Queue Name
    inbound **23**

outbound **28**
MSMQ Receive Action Code
outbound **30**
MSMQ Receive Interval
inbound **24**
MSMQ Share Mode
inbound **23**
outbound **29**
properties editor **20**

R

RECEIVE_ACCESS **30**
rollback
enabling for outbound messages **37**
Runtime Java API
adding **15**
Runtime win32 bridge DLLs
adding **15**

S

SEND_ACCESS **30**
supported operating systems **10**

T

text conventions **8**